Vysoká škola báňská – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Měření přenosových parametrů aplikačních protokolů pomocí přístroje Spirent TestCenter C1

Measuring Transmission Parameters of Application Protocols Using Spirent TestCenter C1

2021 **Jakub Vrlík**

Rád bych poděkoval Ing. Petru Machníkovi, Ph.D. za odbornou pomoc a konzultaci při vytváření této bakalářské práce.

Abstrakt

Bakalářská práce se zabývá měřením a testováním parametrů QoS na různých aplikačních protokolech. První část je věnována popisu kvality služby a jednotlivým parametrům QoS: propustnost, zpoždění, rozptyl zpoždění a ztrátovost paketů. Ve druhé části jsou popsány použité aplikační protokoly: HTTP, FTP a SMTP. Třetí část obsahuje popis použitého generátoru/analyzátoru datového provozu Spirent TestCenter C1 a jeho srovnání s dalšími HW a SW generátory a analyzátory přenosových parametrů. Ve čtvrté části je popsáno měření přenosových parametrů pomocí Spirent TestCenter C1. Poslední část se zabývá mým subjektivním názorem na zařízení Spirent TestCenter C1. Součástí bakalářské práce jsou návody k použití zařízení Spirent TestCenter C1.

Klíčová slova

Spirent, Kvalita služby, Cisco, Aplikační protokoly, HTTP, FTP, SMTP

Abstract

This bachelor thesis deals with measuring and testing parameters of QoS for several application protocols. The first part of thesis deals with description of quality of service and describes individual parameters of QoS: bandwidth, delay, jitter and packet loss. The second part describes application protocols: HTTP, FTP and SMTP, which have been used for testing in part four. The third part describes traffic generator/analyzer Spirent TestCenter C1 and comparison with other HW and SW traffic generators and analyzers. The fourth part deals with measuring transmission parameters using Spirent TestCenter C1. In the last part are described my subjective impressions of Spirent TestCenter C1. Thesis includes instructions for using Spirent TestCenter C1.

Key words

Spirent, Quality of Service, Cisco, Application protocols, HTTP, FTP, SMTP

Seznam použitých symbolů a zkratek

Zkratka	Význam
ARP/NDP	Address Resolution Protocol / Neighbor Discovery Protocol
BER	Bit Error Rate
CoS	Class of Service
DNS	Domain Name System
DuT	Device under Test
ESP	Extreme Scale & Performance
E-mail	Electronic mail
FIN	Finish
FTP	File Transfer Protocol
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ID	Identification Data
IGMPv2	Internet Group Management Protocol version 2
IP	Internet Protocol
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISO/OSI	International Organization of Standardization / Open Systems Interconnection
L2	Second Layer of ISO/OSI model
MSN	Microsoft Network
MTA	Mail Transfer Agent
NAC	Network Access Control
NFS	Network File System
P2P	Peer to Peer
QoS	Quality of Service
RED	Random Early Detection

RFC	Request for Comments
RST	Reset
RTSP/RTP	Real Time Streaming Protocol / Real Time Protocol
SCTP	Stream Control Transmission Protocol
SIP	Session Initiation Protocol
SMTP	Simple Mail Transfer Protocol
SQL	Structured Query Language
SYN/ACK	Synchronize / Acknowledgement
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol / Internet Protocol
UA	User Agent
UDP	User Datagram Protocol
URL	Uniform Resource Locator
VoIP	Voice over Internet Protocol
VLAN ID	Virtual Local Area Network Identification Data
WAN	Wide Area Network
WRED	Weighted Random Early Detection
WWW	World Wide Web

Obsah

Úvod ..... - 9 -
1 Kvalita služby ..... - 10 -
1.1 Parametry kvality služby ..... - 10 -
1.1.1 Propustnost (Bandwidth) ..... - 10 -
1.1.2 Zpoždění (Delay) ..... - 11 -
1.1.3 Rozptyl zpoždění (Jitter) ..... - 12 -
1.1.4 Ztrátovost paketů (Packet loss) ..... - 12 -
2 Základní popis aplikačních protokolů ..... - 13 -
2.1 Hypertext Transfer Protocol ..... - 13 -
2.2 File Transfer Protocol ..... - 14 -
2.3 Simple Mail Transfer Protocol ..... - 15 -
3 Popis analyzátorů a generátorů datového provozu ..... - 16 -
3.1 Spirent TestCenter C1 ..... - 16 -
3.2 FETest Parascope GigE ..... - 16 -
3.3 iPerf ..... - 17 -
3.4 Srovnání ..... - 18 -
4 Testování funkcí zařízení Spirent TestCenter C1 ..... - 19 -
4.1.1 Spirent TestCenter Layer 4-7 Application ..... - 19 -
4.1.2 Spirent TestCenter Layer 4-7 Results Analyzer ..... - 26 -
4.2 Testování protokolu HTTP ..... - 27 -
4.2.1 Nastavení zařízení Spirent a Nginx serveru ..... - 27 -
4.2.2 Testování při maximálně 1 000 transakcí na serveru zařízení Spirent ..... - 29 -
4.2.3 Testování při maximálně 1 000 transakcí na serveru Nginx ..... - 31 -
4.2.4 Testování při maximálně 100 000 transakcí na serveru zařízení Spirent - 32 -
4.2.5 Testování při maximálně 100 000 transakcí na serveru Nginx ..... - 33 -
4.3 Testování protokolu FTP ..... - 35 -
4.3.1 Nastavení zařízení Spirent ..... - 35 -
4.3.2 Testování při maximálně 50 spojeních ..... - 36 -
4.3.3 Testování při maximálně 5 000 spojeních ..... - 38 -

4.4 Testování protokolu SMTP ..... - 40 -
4.4.1 Nastavení zařízení Spirent ..... - 40 -
4.4.2 Testování při maximálně 5 000 simulovaných uživatelů ..... - 41 -
4.4.3 Testování při maximálně 50 000 simulovaných uživatelů ..... - 43 -
4.5 Testování několika aplikačních protokolů (HTTP, FTP, SMTP) ..... - 45 -
4.5.1 Nastavení zařízení Spirent ..... - 45 -
4.5.2 Testování ..... - 46 -
5 Zhodnocení zařízení Spirent TestCenter C1 na základě praktické zkušenosti ..... - 50 -
Závěr ..... - 51 -
Použitá literatura ..... - 52 -
Seznam příloh ..... - 54 -

Úvod

V dnešní době Internetu a počítačových sítí jako takových vznikají nové a nové aplikace. Aby běžní uživatelé mohli tyto aplikace využívat je nutné mít server, který danou službu poskytuje. Ať už se jedná o stream, e-mail nebo běžný web je potřeba kvalitního serveru. Než se server nasadí do běžného provozu je nutné ho otestovat, k tomu využíváme generátory a analyzátory datového provoz mezi které patří například Spirent TestCenter C1. Generátory a analyzátory datového provozu můžeme využívat mimo jiné i na testování počítačových sítí a parametrů přenosu.

První kapitola se zabývá výše zmíněnými parametry přenosu a kvalitou služby. Jsou zde vysvětleny a popsány jednotlivé parametry včetně toho, jak tyto parametry ovlivňujeme v provozu. Druhá kapitola se věnuje aplikačním protokolům použitým v praktické části bakalářské práce. Třetí část se věnuje jednomu ze základních cílům této práce a tím je srovnání několika hardwarových a softwarových generátorů a analyzátorů datového provozu. Čtvrtá kapitola je věnována praktickému testování parametrů provozu. S tím souvisí další cíl bakalářské práce a tím je otestování alespoň 3 funkcí zařízení Spirent TestCenter C1 se zaměřením na protokoly na 4.-7. vrstvě ISO/OSI modelu. Pro lepší pochopení problematiky čtvrté kapitoly doporučuji také přečíst přiložené přílohy ve kterých jsou návody k použití zařízení Spirent TestCenter C1 v rámci generování datového provozu na úrovni aplikační vrstvy. Poslední kapitola se zaobírá posledním úkolem této bakalářské práce a tím je sepsání dojmů z praktického použití přístroje a jeho srovnání s podobnými technologiemi.

1 Kvalita služby

Kvalita služby neboli Quality of Service či zkráceně QoS je jedna ze základních technologií síťové infrastruktury. U technologie QoS rozdělujeme provoz do tříd podle parametrů kvality služby. Mezi tyto parametry patří propustnost, zpoždění, rozptyl zpoždění a ztrátovost paketů. Tyto parametry budou blíže popsány v podkapitole 1.1. Podobným mechanismem je Class of Service, zkráceně CoS. Zde dochází k rozdílnému zacházení s rámci v závislosti na třídě, do které patří. CoS je omezeno na L2 vrstvu a využívá k identifikaci 3 bity, což znamená, že existuje 8 různých tříd.

Hlavní účel QoS je správa síťových zdrojů, přičemž se snaží maximalizovat kvalitu jednotlivých relací. S pakety by se nemělo zacházet rovnocenně, protože si všechny nejsou rovny. Funkce QoS využívají systém řízené nespravedlnosti, to znamená, že některé relace obdrží pakety dřív než jiné. Například relace citlivé na zpoždění jako je třeba telefonní hovor přes VoIP bude mít vyšší prioritu než relace, kde dochází ke komunikaci s webovým serverem pomocí HTTP. Tento systém zajišťujeme pomocí čtyř základních funkcí: plánováním, obsluhou paketových front, omezováním toků a prevencí zahlcení.

QoS se původně v počítačových sítích neřešilo, a proto se zde používala metoda Best Effort, kde nedocházelo k prioritizaci přenášených paketů a v případě zahlcení se pakety jednoduše zahodily. V rámci QoS byl Best Effort nahrazen metodami Integrated Services a Differentiated Services, téže přezdívané jako Hard QoS a Soft QoS. [1][5][6]

1.1 Parametry kvality služby

V rámci QoS máme 4 základní parametry, které ovlivňují kvalitu služby. Tyto parametry se snažíme měnit tak, aby byla výsledná kvalita relace co nejvyšší. Každý parametr má svůj specifický způsob, jak jej lze měnit. Bohužel v praxi nemůžeme nikdy dosáhnout perfektního výsledku, neboť se může stát, že pokud vylepšíme hodnoty jednoho parametru, tak hodnoty jiného parametru se naopak můžou zhoršit. Ing. Petr Grygárek Ph.D. tvrdí, že „cílem mechanismů zajištění QoS je pro datový tok garantovat alespoň určitou minimální šířku pásma, shora omezené zpoždění a minimální rozptyl zpoždění (jitter).“ [1][5][6][7]

1.1.1 Propustnost (Bandwidth)

V telekomunikačních sítích je propustnost omezena nejméně propustnou linkou mezi komunikujícími zařízeními a mírou zatížení síťových prvků na trase. Obecně řečeno se jedná o maximální počet přenesených bitů za sekundu po dané lince. Každá přenosová technologie má jinou maximální propustnost, může se jednat o desítky kbit/s až stovky Gbit/s.

V praxi tento parametr korigujeme takzvaným omezováním toků. K tomu využíváme dvou základních algoritmů: Leaky bucket a Token bucket. [2][6][9]

1.1.2 Zpoždění (Delay)

Jedná se o čas, který je dán trváním přenosu paketu z jednoho koncového bodu do jiného koncového bodu. Síťové zpoždění může způsobovat několik různých faktorů, které se dělí na fixní a variabilní. Fixní zpoždění je způsobeno serializací a propagací.

Zpoždění v telekomunikačních sítích řešíme pomocí takzvaného plánování a s ním související obsluhou paketových front. [1][6]

1.1.2.1 ***Serializační zpoždění***

Čas nutný pro konverzi rámců na elektrické či optické pulzy. Toto zpoždění je fixní a lze jej vypočítat pomocí vzorce uvedeného níže.

$$Serializační\ zpoždění = \frac{Počet\ odeslaných\ bitů\ [bit]}{Přenosová\ rychlost\ linky\ [bps]}$$

Výsledné zpoždění je obvykle v milisekundách. [1][8]

1.1.2.2 ***Propagační zpoždění***

Zpoždění, které nám říká za jak dlouho se dostane bit z jednoho konce spoje na druhý. Nejvýznamnější faktor síťového zpoždění na WAN linkách je takzvaná propagace, může zde zapříčinit přes 95 % síťového zpoždění. Opět se jedná o fixní zpoždění, a proto jej lze ho vypočítat pomocí vzorce:

$$Propagační\ zpoždění = \frac{Délka\ linky\ [m]}{Propagační\ rychlost\ [m/s]}$$

Jako propagační rychlost můžeme použít dvě hodnoty a to jsou: 3*10^8, což je rychlost šíření světla ve vakuu, nebo lze použít i hodnotu 2,1*10^8, která nám dá věrohodnější výsledek, neboť bere do úvahy nejen optická vlákna, ale i metalická vedení. Rychlost 3*10^8 je spíše teoretická a v praxi se využívá hodnota 2,1*10^8. Délka linky je vzdálenost mezi dvěma směrovači. Výsledná hodnota je obvykle v milisekundách. [1][8]

1.1.2.3 ***Zpoždění způsobené obsluhou paketových front***

Variabilní zpoždění způsobené čekáním paketů ve frontách na jednotlivých zařízeních. Pokud sečteme Serializační zpoždění, Propagační zpoždění a zpoždění způsobené obsluhou paketových front získáme hodnotu Network delay neboli síťové zpoždění. [7][8]

1.1.2.4 ***Zpoždění způsobené přesměrováváním***

Zpoždění způsobené dobou předávání přijatých paketů do výstupní fronty. Zpoždění je variabilní a nelze vypočítat pomocí vzorce. V celkovém přenosu je toto zpoždění zanedbatelně nízké a lze jej při výpočtech ignorovat. [8]

1.1.2.5 ***Zpoždění způsobené tvarováním provozu***

Při tvarování provozu (omezování toků) dochází k zpomalení obsluhy paketových front, a tudíž vzniká toto variabilní zpoždění. V závislosti na situaci můžeme zvolit pomalejší odesílání paketů s tím, že dojde k zpoždění z důvodu pomalé obsluhy a tím snížíme

pravděpodobnost zahazování paketů, nebo zvolíme rychlejší odesílání, ale bude zde vyšší náchylnost k zahození paketů. [8]

1.1.3 Rozptyl zpoždění (Jitter)

Tento parametr je velice blízký parametru zpoždění, oba parametry se měří stejně, tudíž mají stejné jednotky. Běžně tyto hodnoty nabývají jednotek až stovek milisekund. Jitter je odchylka zpoždění příchodu mezi sekvenčními pakety. Taktéž je známý jako variabilní zpoždění. Variabilní zpoždění je způsobeno frontizací, přesměrováváním či tvarováním.

Zpoždění eliminujeme plánováním a obsluhou paketových front, podobně jako u parametru Delay jelikož si jsou s Jittrem velice blízké. [1][7][8]

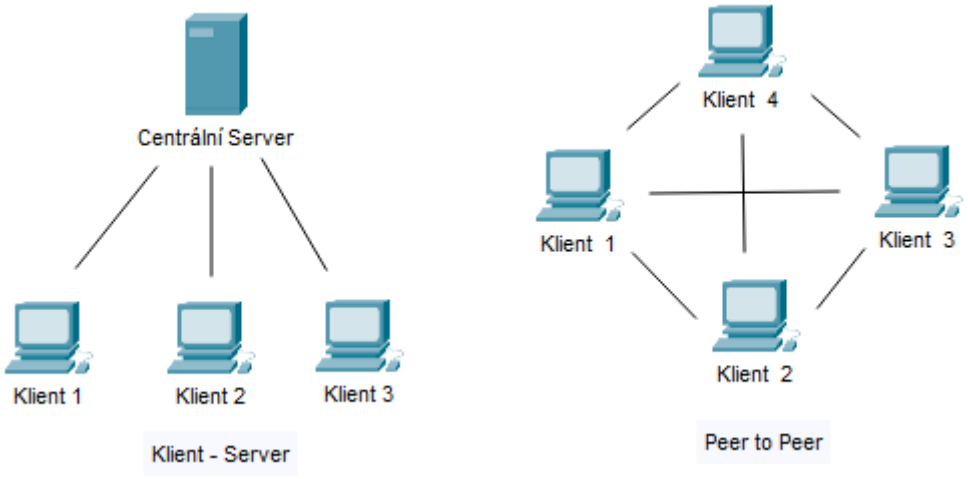
1.1.4 Ztrátovost paketů (Packet loss)

Poslední parametr je takzvaný packet loss. Jedná se o poměr všech odeslaných paketů k celkovému počtu správně přijatých paketů. Tento parametr je uváděný v procentech. Ke ztrácení paketů dochází především díky chybám přenosu na fyzické vrstvě, avšak pakety můžou zahazovat i síťové prvky v závislosti na přetížení procesoru přerušeními od vstupního provozu či přeplnění výstupních front. Ztrátovost paketů nám nejvíce vadí u přenosu hlasu a videa, kde se primárně používá UDP jako transportní protokol a není zde zajištěno opětovné zaslání ztraceného paketu, přesto jsou k ztrátovosti tolerantní, neboť pokud se jedná o malé ztráty, tak je koncový uživatel ani nemusí zaznamenat. Proto chceme u přenosu hlasu a videa co nejnižší ztrátovost. V rámci přenosu dat pomocí TCP nám ztrátovost nevadí, tak jako u protokolu UDP, protože je zde garantováno doručení všech paketů, v případě ztráty se jednoduše zašle znovu, tím se však může zhoršit jiný přenosový parametr (například zpoždění) avšak výsledná data budou kompletní.

Ztrátovost paketů v důsledku zahazování síťovými prvky řešíme prevencí zahlcení a jimi dostupnými metodami jako je RED nebo WRED. [1][4][6]

2 Základní popis aplikačních protokolů

Aplikační služby využívající protokolové rodiny TCP/IP se v počítačových sítích poskytují za pomoci speciálních aplikačních protokolů. Zařízení spolu komunikují prostřednictvím architektury klient-server nebo P2P. Komunikace probíhá za pomoci speciálních aplikačních portů. Aplikační protokoly uvedené v RFC jsou veřejně přístupné, jako například protokol HTTP, takže kdokoliv si může udělat svůj vlastní web, ale jsou zde i jiné protokoly, které jsou patentované a záměrně nedostupné veřejnosti, tyto protokoly smí využívat pouze společnosti, které mají práva na tyto protokoly, mezi tyto protokoly se řadí například protokol Skype.



Obrázek 2.1: Architektura Klient - Server (vlevo) a architektura P2P (vpravo)

Při testování kvality služby se v této práci bude pracovat s protokoly HTTP, FTP a SMTP, které jsou více popsány níže. [3] [4]

2.1 Hypertext Transfer Protocol

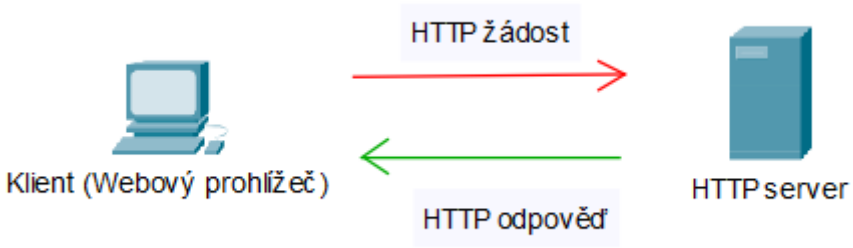
Do roku 1990 se Internet využíval primárně v akademické sféře a mimo ní byl prakticky neznámý. V roce 1990 se na scéně objevila aplikace WWW (World Wide Web), jedná se o první aplikaci, která upoutala širokou veřejnost a způsobila masivní nárůst uživatelů Internetu, čímž zároveň začaly vznikat další aplikaci a pomalu se začal Internet přetvářet do podoby, jak ho známe dnes.

HTTP je aplikační protokol služby Web, je definován v RFC 1945 a RFC 2616. Jako transportní protokol používá TCP a pracuje na portu 80, protokol HTTP má také svou šifrovanou verzi HTTPS, která pracuje na portu 443. Protokol je určený k přístupu a výměně informací v distribuovaných hypermediálních informačních systémech. K přenosu www stránek

využívá formát HTML. HTTP se taktéž používá ke zpřístupňování služeb jiných aplikačních protokolů, jako je například SMTP, FTP a další.

Před zahájením přenosu musí být navázáno spojení pomocí protokolu TCP. Poté dochází k výměně zpráv mezi klientem a serverem. Zprávou "request" klient zasílá serveru své HTTP požadavky, kromě samotného požadavku obsahuje zpráva i záhlaví. V takovéto zprávě nalezneme informace o čase a datu HTTP požadavku, typ požadavku, typ autorizace přístupu k serveru a další. Zprávou "reply" reaguje server na zprávu typu "request". "Reply" opět obsahuje kromě samotné zprávy i záhlaví. V tomto záhlaví jsou informace o verzi HTTP a kód odpovědi. U "reply" zpráv rozlišujeme 5 různých odpovědí: 1xx - Information, 2xx - Success, 3xx - Redirection, 4xx - Client error, 5xx - Server error. [3][4]

V rámci protokolu HTTP máme několik typů žádostí v původní verzi protokolu 1.0 byly definovány celkem 3 základní metody. První metodou je GET, touto metodou žádáme o poskytnutí informací zadané URL adresy. Druhou metodou je HEAD, metoda je totožná s GET s tím rozdílem, že server nevrací tělo zprávy, ale pouze informace obsažené v hlavičce. Poslední metodou je POST, touto metodou odesíláme informace na server. Ve verzi HTTP 1.1 byly definovány další 4 metody. První z nich je PUT, jedná se o obdobu příkazu POST s tím rozdílem, že při několika volání metody PUT vygenerujeme vždy stejný výsledek, při opakovaném volání totožné metody POST vznikají vedlejší účinky, protože vytváříme několikrát stejný zdroj. Další je metoda DELETE, touto metodou žádáme server o smazání požadované URL. Metodou OPTIONS žádáme server o informace týkající se komunikace identifikovaného URL. Poslední metoda je TRACE, kterou zjišťujeme cíl požadavku na aplikační vrstvě. [13][14][15]

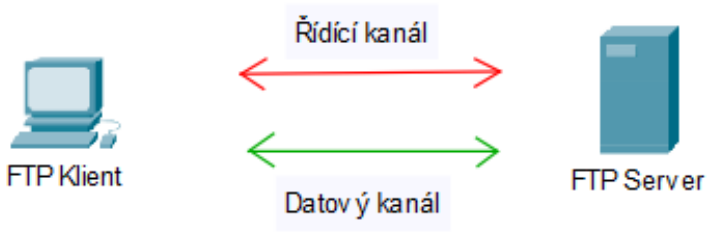


Obrázek 2.2: Komunikace pomocí HTTP mezi klientem a serverem

2.2 File Transfer Protocol

Protokol FTP umožňuje koncovým uživatelům provádět široký výběr operací se soubory a adresáři počítačů. Opět se jedná o veřejně přístupný protokol a v RFC je blíže specifikován pod RFC 959. Jako transportní protokol využívá TCP. Nejčastěji se FTP používá ke sdílení dat, například hudba či videa.

Protokol využívá pro přenos dvě paralelní spojení. První spojení probíhá na portu 21, slouží k řízení spojení a posílají se po něm řídící informace. Druhé spojení probíhá na portu 20 a používá se k samotnému přenosu dat. Před samotným přeposíláním dat se uživatel musí pomocí svého ID a hesla autentizovat a následně autorizovat. Poté může libovolně posílat soubory z vzdáleného počítače/serveru na svůj lokální počítač/server a naopak. [3][4]



Obrázek 2.3: Komunikace pomocí FTP mezi klientem a serverem

2.3 Simple Mail Transfer Protocol

Poslední aplikační protokol použitý v této práci je SMTP. Využívá se pro přenos e-mailů (elektronické pošty). SMTP komunikuje na portu 25 za pomoci koncových uzlů (UA) a poštovních serverů (MTA). Na transportní vrstvě opět využívá protokol TCP. Taktéž je specifikován v RFC pod označením RFC 821.

SMTP funguje pomocí 5 základních příkazů: HELO slouží k vzájemnému představení komunikujících SMTP serverů a navázání spojení, MAIL FROM - tímto příkazem říkáme, kdo je odesílatelem e-mailu, MAIL TO - zde je adresa příjemce e-mailu, DATA - za tímto příkazem následuje odesílaná zpráva a pomocí příkazu QUIT se ukončí spojení. Přičemž cílový server odpovídá na každý příkaz. [3][4]

```
S: 220 smtp2go.com ESMTP Exim
C: HELO mydomain.com
S: 250 Hello mydomain.com
C: MAIL FROM:<sender@mydomain.com>
S: 250 Ok
C: RCPT TO:<recipient@anotherdomain.com>
S: 250 Accepted
C: DATA
S: 354 Enter message, ending with "." on a line by itself
C: Subject: sample message
C: From: sender@mydomain.com
C: To: recipient@anotherdomain.com
C:
C: Greetings,
C: Typed message (content)
C: Goodbye.
C: .
S: 250 OK
C: QUIT
S: 221 www.sample.com closing connection
```

Obrázek 2.4: Komunikace pomocí SMTP protokolu [16]

3 Popis analyzátorů a generátorů datového provozu

Tato kapitola bude primárně zaměřena na popis jednotlivých vlastností použitého generátoru datového provozu Spirent TestCenter C1. Mimo jiné zde bude popsán další hardwarový a softwarový generátor datového provozu a na konci kapitoly bude jejich srovnání.

3.1 Spirent TestCenter C1

Generátor/analyzátor datového provozu umožňující realistické testování mobilních, pevných a cloudových infrastruktur, cloudových aplikací, datových center, sítí WAN či podnikových areálů. Pro testování aplikací realisticky napodobuje chování vrstev L4-L7. Kromě toho generuje realistický provoz na vrstvách L2 a L3, kde testujeme mechanismy QoS.

V rámci aplikačních protokolů lze generovat široké portfolio aplikačních protokolů od datových, jako je HTTP, DNS, SMTP, FTP, přes hlasové protokoly, jako je protokol SIP a taktéž protokoly pro přenos videa, jako jsou RTSP/RTP či IGMPv2. Dále podporuje autentizační protokoly: 802.1x, Radius a NAC. Kromě těchto protokolů taktéž podporuje některé rozšířené protokoly jako jsou: BitTorrent, SQL, Gnutella, Yahoo, SKYPE, Oracle, MSN či NFS. Co se týče analýzy na vrstvách L2 a L3, přístroj provádí přes 35 měření v reálném čase.

Přístroj je velký 88,9x330,2x254 mm, je nutné jej napájet ze zdroje o 100-240 V střídavého napětí. Obsahuje 4 10/100/1000 Baset-T porty, dvě optické rozhraní: 10 GBASE-SR (850 nm) pro multimode vlákna a 10 GBASE-LR (1310 nm) pro singlemode vlákna, 4x 10/5/2,5/1G/100M porty. K obsluze zařízení je nutný počítač se systémem Windows. [10]

Obrázek 3.1: Zařízení Spirent TestCenter C1 [10]

3.2 FETest Parascope GigE

Je generátor a zároveň analyzátor datového provozu. Umožňuje uživateli nastavit vlastní profily pro testování a tím výrazně urychlí celkové testování. Poskytuje pokročilé diagnostické nástroje pro Ethernet, jako je Ping, TraceRoute, FTP, IPDV, Blinking lokátor,

měřič délky kabelu, měřič optického výkonu či test mapy vodičů. Umožňuje testovat BER na L1-L3, referenční testování RFC 2544 zabývající se propustností, ztrátovostí rámců, zpožděním. Kromě toho dokáže detekovat MAC/IP kolize, generovat a analyzovat provoz v rámci QoS služeb. Taktéž podporuje testování aplikační vrstvy a jí používané protokoly, jako je DNS, POP3, SMTP či HTTP. Přístroj taktéž umožňuje vzdálené ovládání.

Samotné zařízení je velké 190x108x50 mm a váží 1.1 kg. Zařízení obsahuje baterii s velikostí 5000 mA, která umožňuje uživateli používat přístroj až 7h bez nabití. Přístroj obsahuje 2 10/100/1000 Base-T a 2 100/1000 Base-X ethernetové porty. Kromě ethernetových portů má také 2x USB 1.1, 1x RS-232 a 1x 10 Base-T management port. [11]

Obrázek 3.2: Zařízení FETest Parascope GigE [11]

3.3 iPerf

Je softwarový diagnostický nástroj podporující nastavení různých parametrů protokolů TCP, UDP, SCTP, IPv4 a IPv6. Během testů uvádí QoS parametry jako je například bandwidth. Samotný software je cross-platformový, což znamená, že jej lze použít na jakémkoliv podporovaném systému, mezi které patří například: Linux, Windows, Android, MacOS X. Software je volně ke stažení na oficiální stránkách iPerfu.

V rámci diagnostiky provozu pracujícího s protokoly TCP a SCTP je uživateli umožněno měřit šířku pásma. Při měření provozu s protokolem UDP měří ztrátovost paketů, zpoždění a rozptyl zpoždění. [12]

3.4 Srovnání

Z hlediska přenositelnosti je na tom nejlépe FETest Parascope GigE, přístroj je vcelku malý a nepotřebuje žádný napájecí zdroj, oproti iPerfu, který potřebuje ke svému provozu notebook či počítač, přístroj od Spirentu zde není potřeba zmiňovat, neboť je docela velký, a ještě je potřeba počítače nebo notebooku k jeho provozu.

Co se týče cenové dostupnosti, zde vítězí iPerf, neboť je zdarma, FETest Parascope GigE je starší a podporuje méně funkcí než Spirent TestCenter C1, zařízení od Spirentu je z těchto řešení nejdražší.

Z pohledu generování datového provozu vede Spirent TestCenter C1, které jich podle datasheetu umožňuje nejvíc, jako druhý je poté FETest Parascope GigE a nejhůře odchází iPerf.

Z pohledu analýzy datového provozu jsou na tom přístroje podobně jako u generování.

Pro detailnější měření je nejvhodnější zařízení od Spirentu, pro měření v terénu je naopak vhodný přístroj od FETestu, software iPerf je naopak vhodný pro jednorázové užití, neboť uživatel nemusí platit za jeho použití. V celkovém důsledku má každé zařízení své pro a proti a je jen na koncovém uživateli, které vlastnosti jsou pro něho důležité.

4 Testování funkcí zařízení Spirent TestCenter C1

V této kapitole bude popsána funkčnost jednotlivých aplikací dostupných k zařízení Spirent TestCenter C1 a generování datového provozu se zaměřením na aplikační protokoly. Software zařízení Spirent TestCenter C1

Součástí zařízení Spirent TestCenter C1 je příslušný software, který obsahuje 6 aplikací (Spirent TestCenter Application, Spirent TestCenter Layer 4-7 Application, Avalanche Attack Designer, Spirent TestCenter Session Manager, Spirent TestCenter Results Reporter a Spirent TestCenter Layer 4-7 Results Analyzer). V rámci řešení této bakalářské práce se používal primárně Spirent TestCenter Layer 4-7 a Spirent TestCenter Layer 4-7 Results Analyzer.

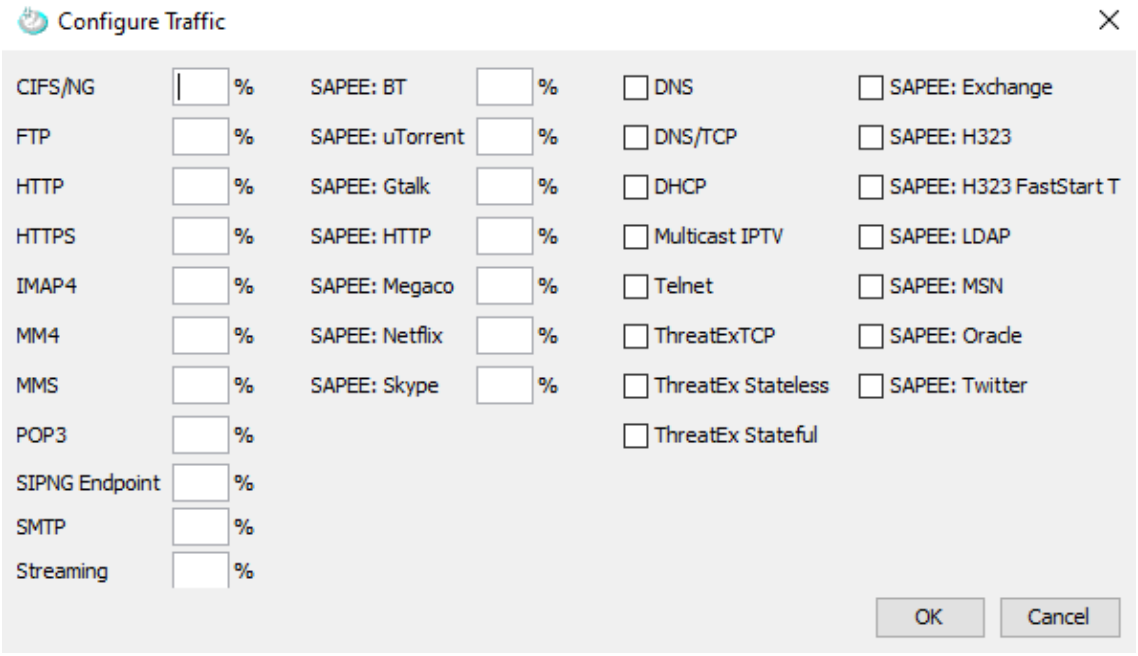
4.1.1 Spirent TestCenter Layer 4-7 Application

Tento program je určen pro generování a testování provozu s důrazem na aplikační protokoly. Lze zde měnit různé parametry aplikací a nastavovat různé typy testování. Lze buďto pouze generovat provoz, což znamená, že program simuluje pouze stranu klienta, nebo je možné simulovat jak stranu serveru, tak stranu klienta. Oba typy umožňují 2 různé testy: Quick test a Advanced test, s tím, že při simulaci klienta i serveru je zde třetí možnost: EZ test.

4.1.1.1 ***EZ test***

Nejjednodušší z testů, lze zde nastavit pouze základní parametry, jako jsou porty koncových zařízení, IP adresy klientů a adresy serverů, doba trvání generování paketů v ustáleném stavu, propustnost v ustáleném stavu a rozložení provozu jednotlivých aplikací. Samotný test má 4 záložky: Endpoints, Network, Run a Results.

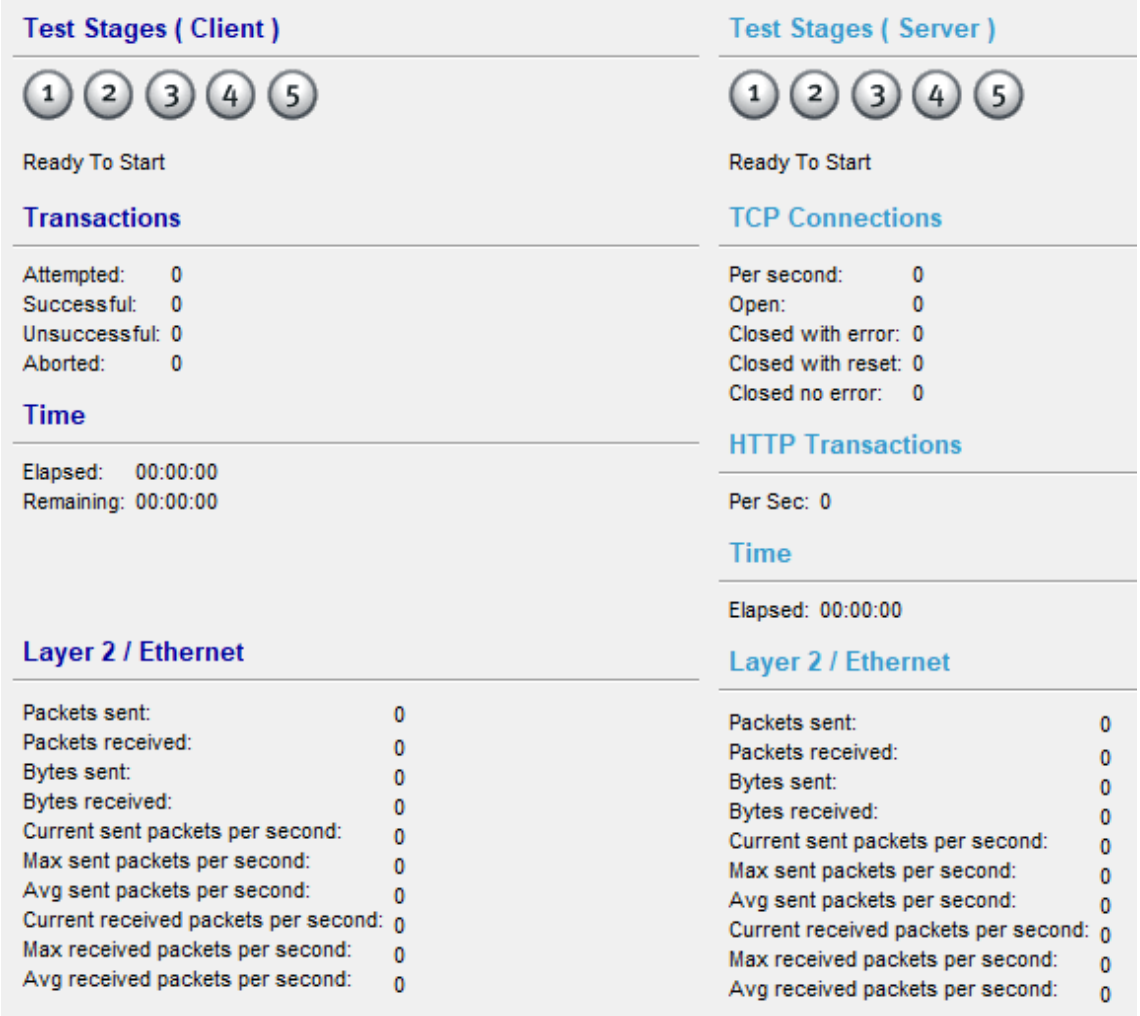
V záložce Endpoints přidělujeme porty zařízení Spirent podle jejich účelu (klient/server), následně lze zvolit chování zasílání paketů, buďto komunikují klienti napřímo se serverem pomocí nastavení ONE-TO-ONE, nebo komunikuje každý klient s každým serverem pomocí nastavení BACKBONE. Lze nastavit i vlastní kombinaci pomocí speciálního profilu. Pokud jsme nezvolili nastavení BACKBONE, tak musíme ještě nastavit jednotlivé spojení. Následně volíme odesílanou zátěž, u tohoto testu je možné definovat zátěž pouze na základě propustnosti, kde je minimum 50000 kbit/s. Lze zvolit i dobu trvání tohoto zatížení, minimální doba je 180 sekund. Nakonec zde můžeme nastavit rozložení provozu jednotlivých aplikací, celkem tento provoz musí být 100 %, jinak se test nespustí.



Obrázek 4.1: Nastavení rozložení provozu podle jednotlivých aplikací

Další záložkou je Network, zde nastavujeme simulované klienty/servery. Přidělujeme jim zde IP adresy, masky, případně i výchozí brány a VLAN ID.

Záložka Run slouží k monitorování generovaného provozu v reálném čase, a to jak na straně klienta, tak na straně serveru. Je zde zobrazeno, v jaké fázi se test nachází, každý EZ test má celkem 5 fází: příprava testu, navyšování provozu, ustálený stav, klesání provozu, test ukončen. Dále se zde zobrazuje stav transakcí na straně klienta, celkový počet pokusů a počet úspěšných, neúspěšných a přerušených transakcí. Na straně serveru je zde stav TCP spojení a jejich aktivity, počet spojení za sekundu, počet otevřených spojení a počet ukončených spojení s chybou, s resetem či bez chyby. Níže nalezneme čas, kolik uběhlo od začátku testu a kolik zbývá do konce. Na straně serveru je zachytáván počet HTTP transakcí za sekundu. Poslední část se týká statistik nižších vrstev ISO/OSI modelu, jde například o počet odeslaných paketů, počet přijatých paketů, maximální a průměrný počet odeslaných/přijatých paketů za sekundu. Je zde možnost zobrazení měřiče přenosové rychlosti na straně klienta, a to jak v příchozím, tak odchozím směru. Kromě výše zmíněných statistik je možné otevřít záznam událostí tlačítkem "Event logs". Tlačítkem Run-Time Stats otevřeme run-time statistiky v grafickém zobrazení.



Obrázek 4.2: Záložka Run

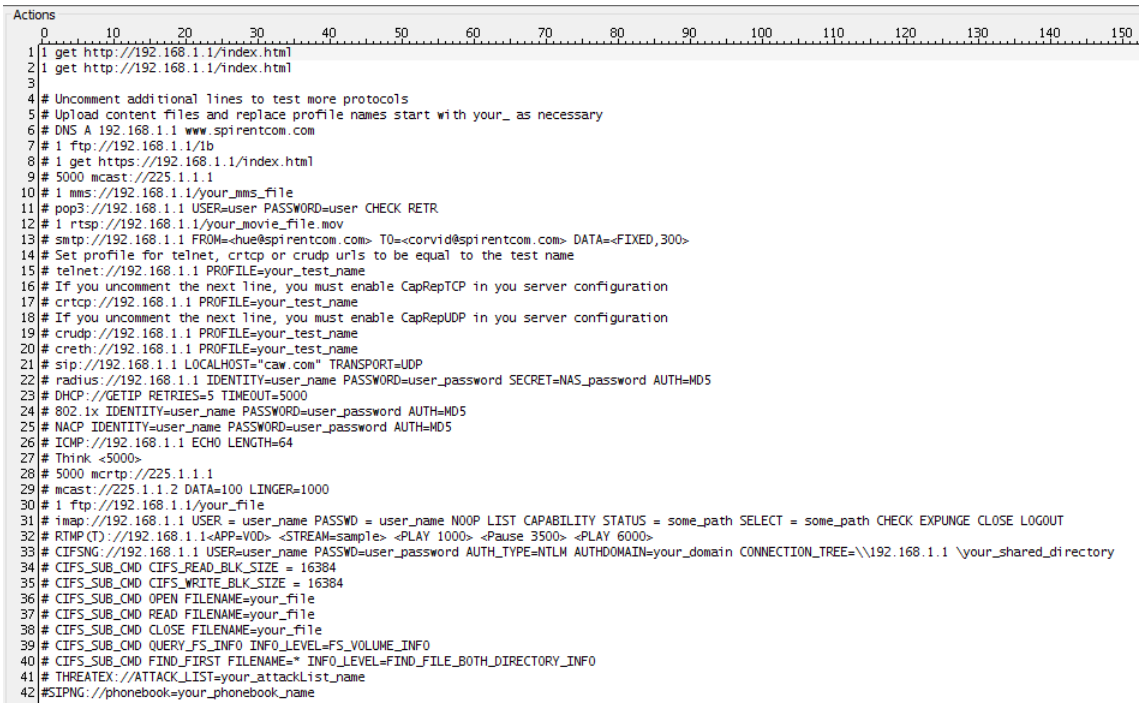
Poslední záložkou je Results. V této záložce jsou jednotlivé testy zobrazeny v tabulce, v každém řádku tabulky je název testu, datum a čas kdy byl test proveden a další 3 parametry podle preferencí uživatele, může mezi ně patřit například celkový počet pokusů transakcí a počet úspěšných transakcí. Každý lze detailně zobrazit přes menu níže, nebo lze později otevřít v programu Spirent TestCenter Layer 4-7 Results Analyzer.

4.1.1.2 ***Quick test***

Oproti EZ testu je komplexnější a umožňuje více funkcí a nastavení, test má celkem 5 záložek: Client, Server, Notes, Monitor, Results.

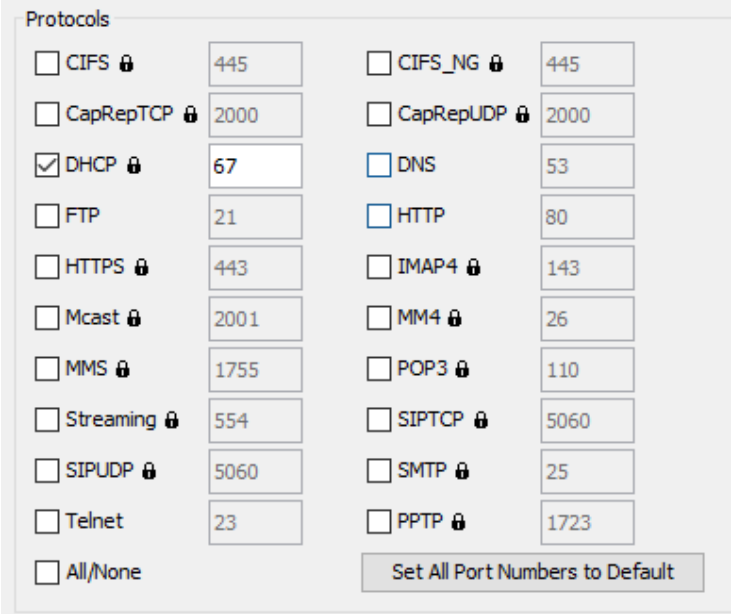
V záložce Client se nachází dvě okna, v prvním je seznam akcí, při vytvoření testu jsou zde už některé akce předvytvořené, pokud je chceme generovat je nutné odebrat znak # před akcí. Do tohoto seznamu je možné přidávat vlastní akce, avšak musí se jednat o reálné příkazy. Další možností je importovat předem připravený seznam akci. Ve druhém okně zadáváme rozsah IP adres, pod jakými bude provoz generován včetně volitelného zadání výchozí brány. Dále je zde nastavení generovaného provozu, lze zde do jisté míry nastavit průběh

generovaných dat, maximální možné zatížení, které můžeme definovat buďto propustností, počtem simulovaných uživatelů, počtem transakcí, počtem spojení a dalšími. Taktéž je zde nastavení délky testu v sekundách, minutách či hodinách. Oproti EZ testu zde není žádná spodní hranice. Nakonec je zde pole pro přidělení portů straně klienta.



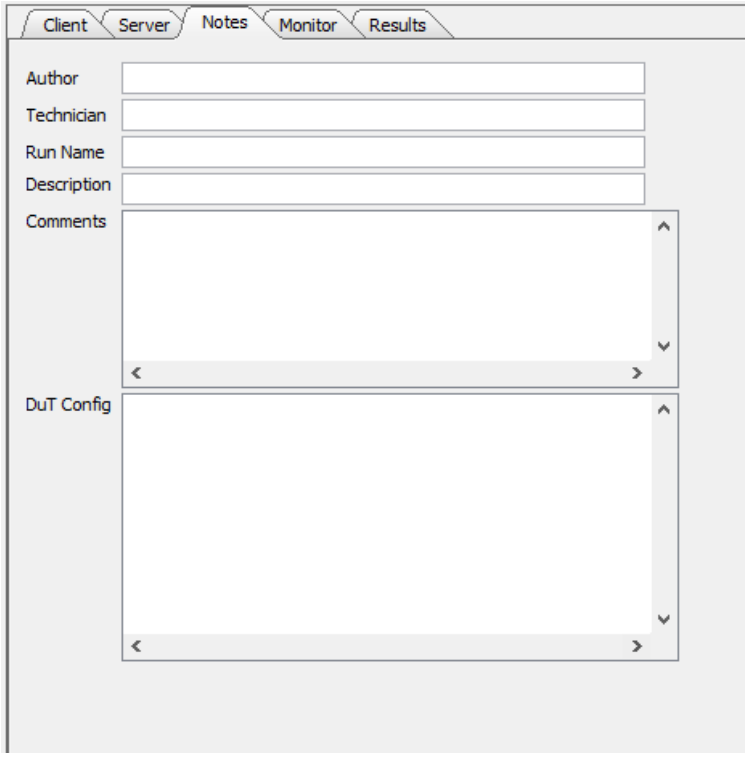
Obrázek 4.3: Záložka Client - okno Actions

Záložka Server slouží ke konfiguraci simulované strany serveru. Je zde nabídka protokolů, které server umí simulovat. V základní nabídce jsou dostupné pouze protokoly DNS, FTP, HTTP a Telnet, pro použití zbylých protokolů je nutná speciální licence. Samotný server může naslouchat i na jiných portech než na těch, které jsou standardní pro daný protokol. Opět lze vložit rozsah IP adres, na kterých bude server. Může se jednat i o jedinou IP adresu. Opět je zde volitelné pole s výchozí bránou a podobně, jako na straně klienta je nutno přiřadit porty straně serveru.



Obrázek 4.4: Záložka Server - volba protokolů

Notes obsahuje políčka: Author - jméno autora testu, Technician - jméno technika spojeného s testem, Run Name - název testu, Description - popisek testu, Comments - komentáře k testu, DuT Config - zde se vkládá informace ohledně konfigurace DuT zařízení (device under test). Záložku Notes není nutné vyplnit. Slouží k specifikaci informací ohledně testu, tyto informace jsou následně uvedeny ve výsledném protokolu.



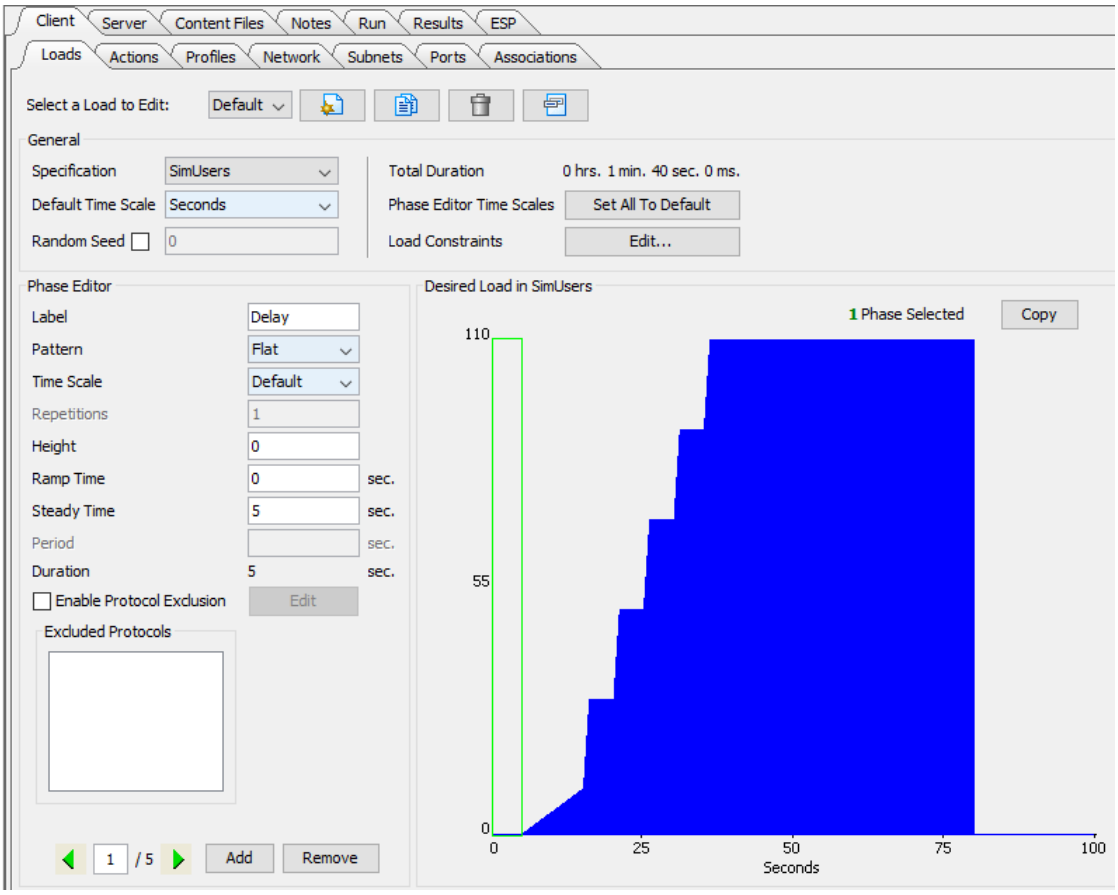
Obrázek 4.5: Záložka Notes

Záložka Results je totožná s tou u EZ testu, záložka Monitor je totožná se záložkou Run u EZ testu.

4.1.1.3 Advanced test

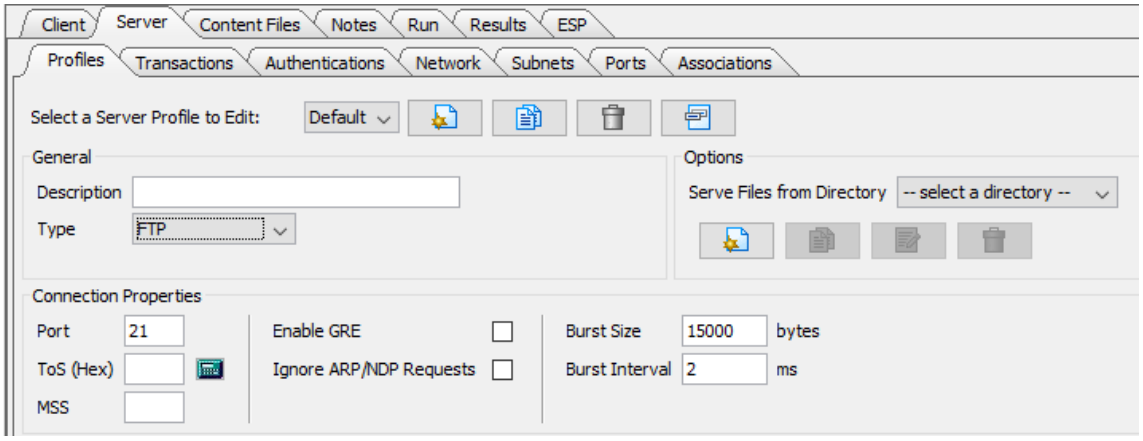
V tomto testu je možné využít veškeré možné prostředky, které Generátor nabízí v rámci vyšších vrstev ISO/OSI modelu. Test je podobný Quick testu s tím, že umožňuje podrobnější nastavení jak na straně klienta, tak na straně serveru. Obsahuje záložky: Client, Server, Content Files, Notes, Run, Results a ESP. Záložky Notes a Results jsou totožné s těmi u Quick testu. Záložky Client a Server mají po 7 podzáložkách. Obě záložky mají podzáložky Network, Subnets, Ports, Associations, které v obou případech plní stejný účel.

První podzáložkou záložky Client je Loads, zde detailně nastavujeme provoz, zejména jeho tvar a dobu trvání testu. Další podzáložkou je Actions, kde podobně jako u Quick testu je seznam akcí, který můžeme měnit. V podzáložce Profiles nastavujeme chování klienta v závislosti na používané službě.



Obrázek 4.6: Záložka Client - podzáložka Loads

V podzáložce Profiles u záložky Server nastavujeme parametry serveru pro jednotlivé služby. Podzáložka Transactions umožňuje nastavení HTTP transakcí, jsou zde pole pro nastavení vlastností odpovědí, hlavičky odpovědí a nastavení adaptivního streamovacího serveru. Podzáložka Authentications slouží k vytvoření účtů, se kterými se klient autentizuje.

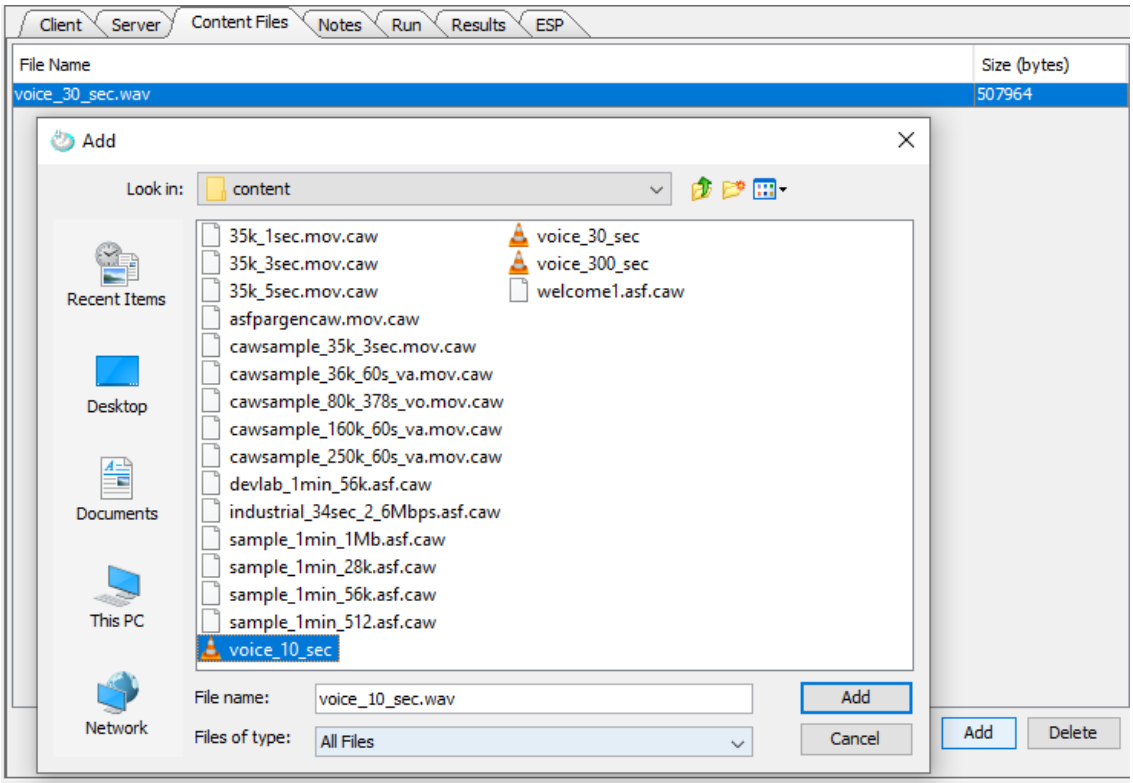


Obrázek 4.7: Záložka Server - podzáložka Profiles

U každé výše zmíněné podzáložky je možné vytvářet speciální profily, které lze vzájemně spojovat v podzáložce Associations.

V podzáložce Network nastavujeme parametry provozu v rámci nižších vrstev protokolového modelu ISO/OSI. Podzáložka Subnets slouží k přidávání jednotlivých podsítí ke klientu/serveru, podzáložkou Ports přidáváme porty zařízení ke klientu/serveru. V podzáložce Associations spojujeme jednotlivé profily s podsítěmi a porty.

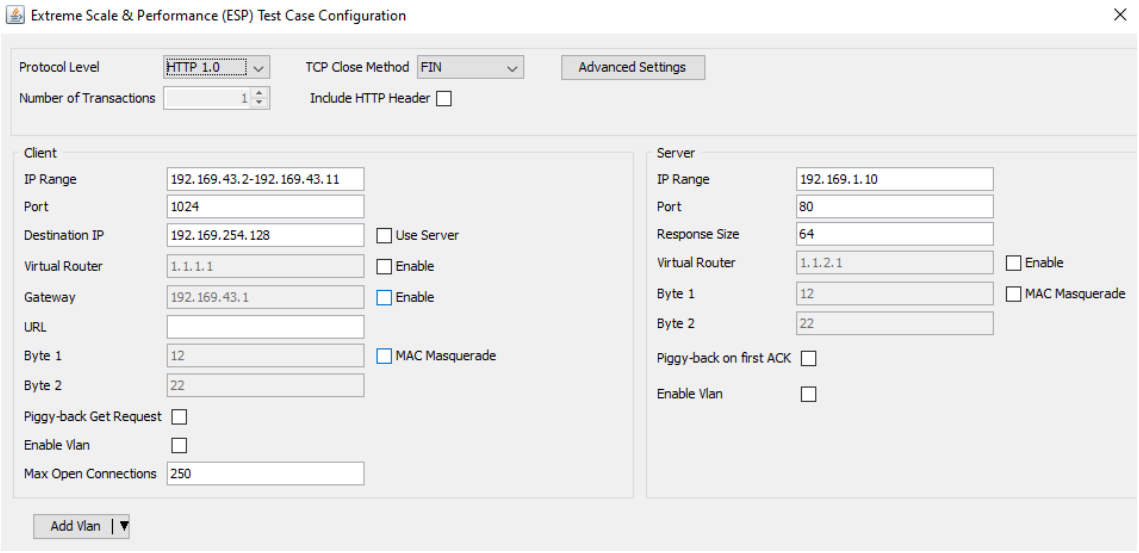
V záložce Content Files můžeme naimportovat soubory, které chceme použít v rámci testu. Může se jednat například o audio záznam hlasu, který chceme přenášet.



Obrázek 4.8: Záložka Content Files

Záložka Run obsahuje celkem 3 podzáložky, první je podzáložka Configure, kde lze nastavit co vše chceme monitorovat. Druhou podzáložkou je Monitor, která je shodná se záložkou Monitor u Quick testu. Poslední podzáložka Load zobrazuje průběh požadovaného i reálného provozu v reálném čase.

Poslední záložkou je ESP - Extreme Scale & Performance, jedná se o funkci, která umožňuje generovat vysoce výkonný síťový provoz zároveň s běžným provozem nakonfigurovaným v předchozích záložkách. ESP umožňuje pouze HTTP spojení a minimální podporovaný port musí být 10 GigabitEthernet.



Obrázek 4.9: Konfigurace ESP

4.1.1.4 ***Srovnání***

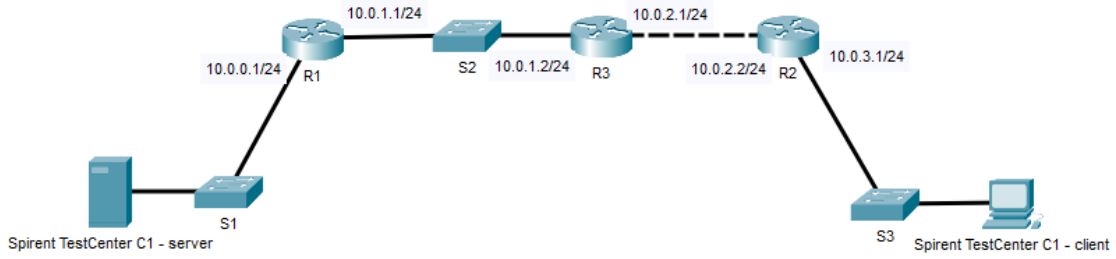
Pro jednoduché testování bez znalostí rozšířených možností zařízení Spirent TestCenter C1 je nejvhodnější EZ test díky jeho přehlednému a jednoduchému rozhraní. Quick test je vhodný pro rychlé odzkoušení funkčnosti reálného serveru, umožňuje totiž simulovat stranu klienta bez nutnosti simulace serveru pomocí zařízení Spirent. Konfigurace samotného testu je lehce složitější než u EZ testu. Oba tyto testy lze kdykoliv konvertovat na Advanced test, opačně to však nelze. Advanced test umožňuje pokročilé nastavení obou simulovaných stran, umožňuje vytvářet spojení mezi jednotlivými klienty a servery, u Quick testu lze definovat spojení pouze obecně. Pro efektivní obsluhu Advanced testu je nutné mít pokročilé znalosti v oblasti aplikačních protokolů.

4.1.2 Spirent TestCenter Layer 4-7 Results Analyzer

Program určený k výsledné analýze zachycených hodnot. V programu je možno otevřít celý adresář s výsledky ve kterém následně vyhledává jednotlivé .csv soubory. Data v nich uložené vkládá do tabulek, ze kterých následně vytváří grafy. Je také možné otevřít libovolné .csv soubory jednotlivě. Tudíž není nutné přenášet všechny výsledky, ale pouze ty, které nás zajímají.

4.2 Testování protokolu HTTP

Veškeré testování probíhalo na topologii, která je zobrazena na obrázku 4.10. Byly použity zařízení značky Cisco. Dvakrát přepínač Catalyst 2950 Series (S1,S2), jeden přepínač Catalyst 3560 Series (S3) a třikrát směrovač 2800 Series. Na propojení byly použity UTP kabely kategorie 5e s maximální přenosovou rychlostí 1 Gbit/s. Propustnost sítě však byla limitována maximální přenosovou rychlostí rozhraní jednotlivých síťových prvků (100Mbit /s).



Obrázek 4.10: Testovací síť

Na směrovačích byly pouze přiděleny IP adresy jednotlivým rozhraním podle obrázku, které byly následně aktivovány. Vzdálené sítě byly do směrovacích tabulek přidány staticky. Na přepínačích byla ponechána výchozí konfigurace. Samotný server byl v síti pod IP adresou 10.0.0.3, strana klienta měla IP adresy v rozsahu 10.0.3.2-10.0.3.9. V rámci HTTP byly provedeny celkem 4 měření, z toho byly 2 na reálném serveru a 2 na simulovaném serveru zařízení Spirent TestCenter C1.

Ve všech měřeních jsou hodnoty v příchozím/odchozím směru vždy vztaženy ke straně serveru.

4.2.1 Nastavení zařízení Spirent a Nginx serveru

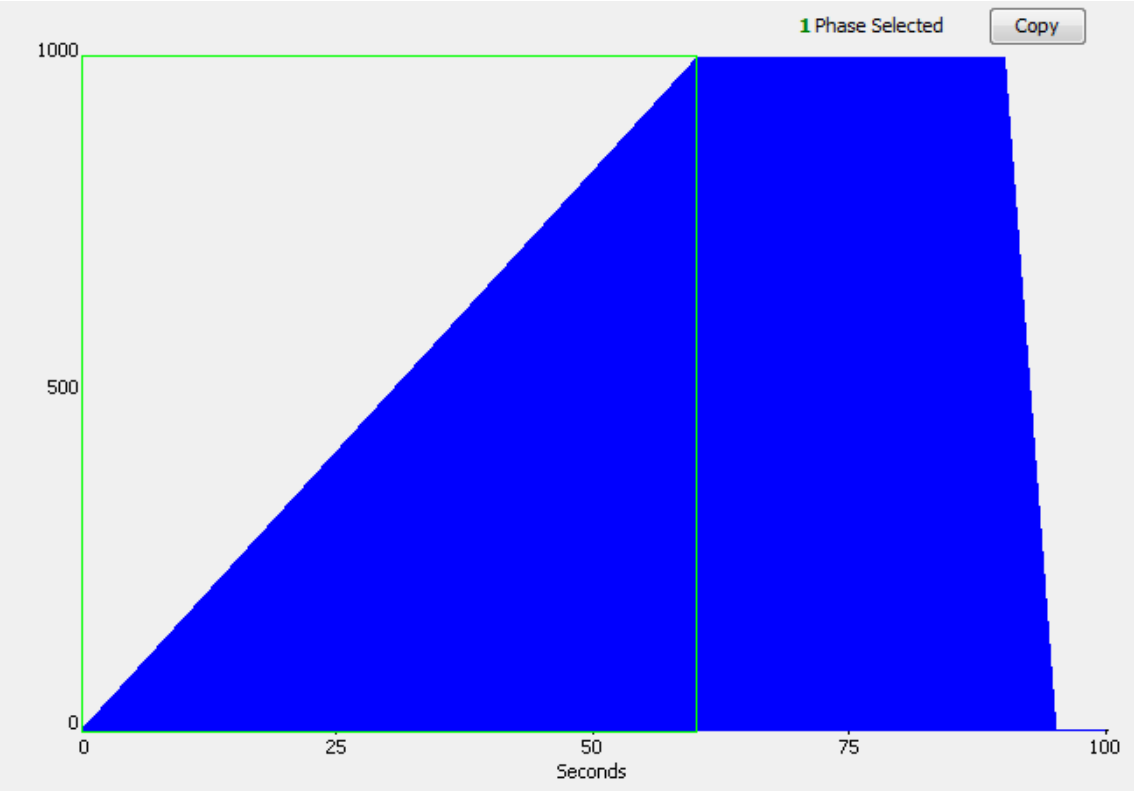
Reálný server Nginx byl nainstalován na stroji s operačním systémem Linux - Ubuntu, jelikož je server určen pouze k testovacím účelům, tak na něm byla ponechána výchozí konfigurace. Server byl pouze nainstalován a následně spuštěn následujícími příkazy.

```
apt-get install nginx //instalace http serveru nginx
apt-get update //aktualizace všech balíčků nainstalovaných na serveru
systemctl start nginx //zapnutí http serveru
systemctl status nginx //ověření stavu serveru
```

Server na Spirentu byl nastaven na HTTP a na verzi HTTP/1.1, taktéž jsem vypnul ignorování ARP/NDP žádostí.

Jednotlivé testy měly totožné nastavení s tím, že se měnil maximální počet generovaných transakcí, tvar samotného provozu můžeme vidět na obrázku 4.11. Při komunikaci mezi klientem a serverem v rámci protokolu HTTP/HTTPS musí být navázáno spojení a poté spolu komunikují pomocí transakcí (žádost na straně klienta, odpověď na straně

serveru) nakonec musí být spojení ukončeno. Pokud definujeme zátěž pomocí transakcí, tak to znamená, že generátor generuje dostatečně velkou zátěž tak aby dosáhl požadované velikosti a následně udržuje daný počet transakcí podle tvaru zátěže.



Obrázek 4.11: Tvar zátěže - 1 000 transakcí

Byly generovány 2 typy akcí. První byla metoda GET, nejzákladnější dotaz v rámci HTTP komunikace, klient tímto žádá server například o zobrazení webové stránky. Druhou akcí byl dotaz HEAD, server vrací pouze hlavičku dané stránky. Pro server simulovaný přístrojem Spirent byla použita následující syntaxe.

```
1 get http://10.0.0.3/index.html
```

```
1 head http://10.0.0.3/index.html
```

Pro Nginx server bylo nutné odebrat "index.html", neboť tato stránka nebyla nakonfigurována, proto výsledné akce vypadaly následovně.

```
1 get http://10.0.0.3/
```

```
1 head http://10.0.0.3/
```

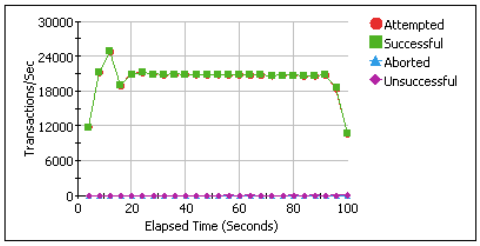
Typ prohlížeče klienta byl nastaven na Chrome. Verze protokolu byla HTTP/1.1 s tím, že mód přenosu byl Persistance, což znamená, že se během jednoho TCP spojení může obsloužit více žádostí. Zároveň bylo aktivované HTTP Pipelining, kde dochází k paralelnímu zasílání žádostí/odpovědí.

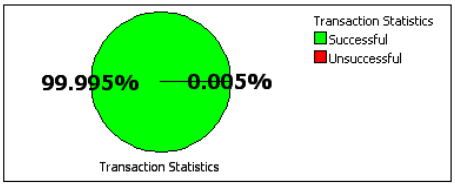
Uživatel byl nastaven tak, že ukončuje spojení hned jak to bude možné a kontroluje délky jednotlivých zpráv, čas přemýšlení uživatele byl nastaven na 5 sekund. Zbylé nastavení na straně klienta i serveru bylo ponecháno ve výchozím nastavení.

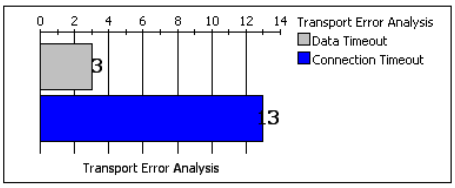
4.2.2 Testování při maximálně 1 000 transakcí na serveru zařízení Spirent

První testování proběhlo na serveru simulovaném pomocí zařízení Spirent. Na obrázku 4.12 lze vidět jednotlivé statistiky přenosu. Samotná zátěž nebyla příliš vysoká, a tudíž úspěšnost provedených transakcí byla téměř 100 %. V první tabulce jsou zobrazeny souhrnné výsledky testu, v první části vidíme celkový počet transakcí, počet úspěšných, neúspěšných a zahozených transakcí, ve vedlejším sloupci jsou tyto počty vztažené k jedné vteřině. Dále je zde zastoupeno zpoždění, avšak ne tak jak je popsáno v teorii o kvalitě služby, ale zpoždění jednotlivých interakcí. Zpoždění jsou v rozsahu od 0 ms po tisíce ms. Přičemž průměrné zpoždění jednotlivých interakcí je v jednotkách až desítkách milisekund. Můžeme si povšimnout například, že zpoždění u prvního přeneseného bytu je větší než zpoždění u již navázaného spojení. V poslední části tabulky lze vidět počet pokusů o navázání spojení a počet navázaných spojení. V druhé tabulce lze vidět parametry, kterými bylo definováno chování klienta. V poslední tabulce jsou více rozepsané statistiky transakcí z první tabulky.

Avalanche
Load Specification: Transactions
Start Time: 2021/04/10 11:32:05
End Time: 2021/04/10 11:33:45
Actual Test Duration: 100 seconds
Protocols: HTTP







Test Results Summary	Transactions			Time (ms)					TCP Connections	
		Total	Rate Per Second		Page Response	URL Response	To TCP SYN/ACK	To First Data Byte	Est. Server Response		Total
	Attempted	2007980	20079	Minimum	0.0	0.0	0.408	0.631	0.0	Attempted	1010920
	Successful	2007879	20078	Maximum	6003.0	6003.0	6004.748	6003.343	4000.926	Established	1010826
	Unsuccessful	101	1	Average	11.13	11.13	41.713	11.129	8.215		
	Aborted	0	0								

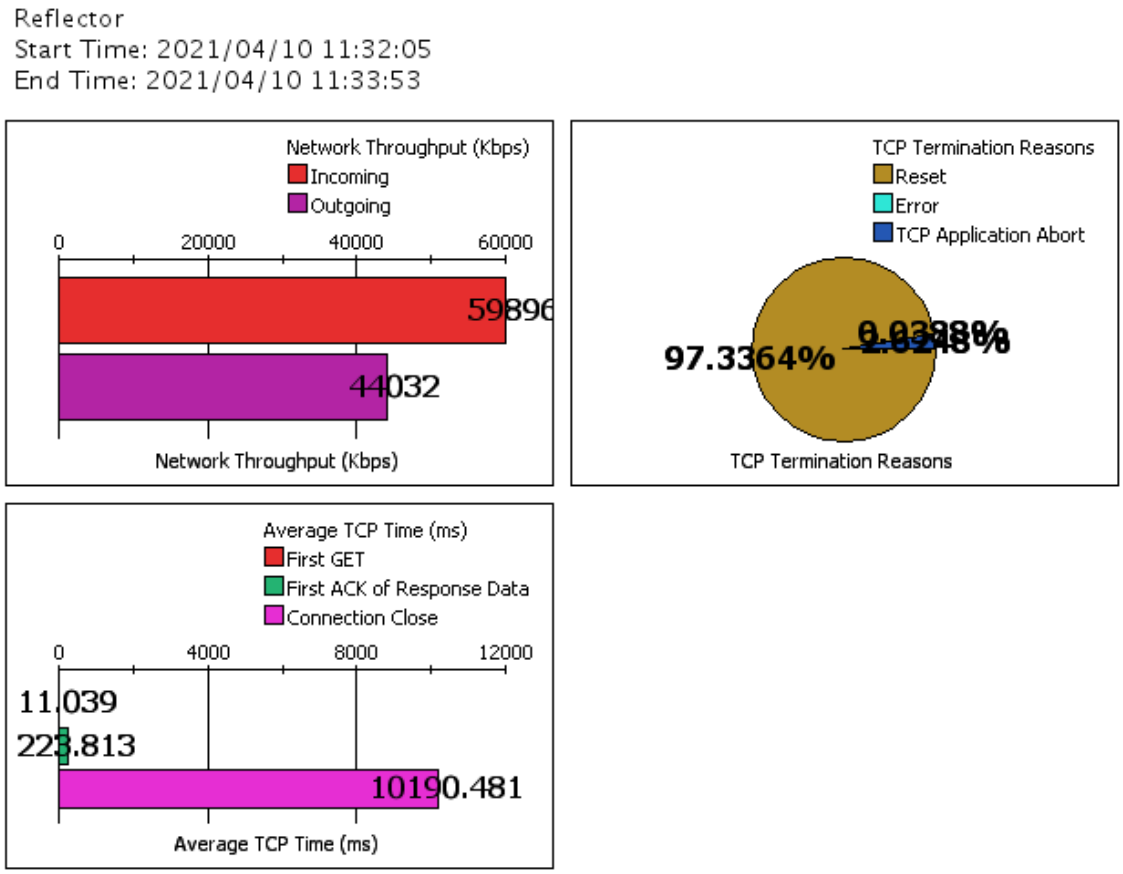
User Profile Definitions	Test							
	Profile	Percentage	Think Time (seconds)	Aborted	Protocol	Persistent Connection	Transactions Per Connection	Connections Per Server
	Default_0	100.000%	5.0	0 % after 0 seconds	HTTP 1.1	Enabled	10	4

Transaction Summary	Test	Count	Transactions (Sub-Commands included)								Response Time (us)		
	Profile	URL	Average Successful Per Second	Attempted	Successful	Unsuccessful	Aborted	Percent Successful	Percent Unsuccessful	Percent Aborted	Minimum	Maximum	Average
	Default_0	2	20078	2007980	2007879	101	0	99.99	0.0	0.0	632.0	6003345.0	11131.4
	Totals	2		2007980	2007879	101	0	99.99	0.0	0.0			

Obrázek 4.12: Souhrnná statistika testování při maximálně 1 000 transakcí (zařízení Spirent jako Klient i Server)

Pokud zařízení Spirent použijeme k simulaci obou stran, tak samotný test vygeneruje i statistiky na straně serveru. Na obrázku 4.13 lze vidět grafy s propustností sítě, a to jak v

odchozím, tak v příchozím směru, důvody ukončení TCP spojení a průměrné zpoždění jednotlivých TCP příznaků. Propustnost sítě byla v tomto měření v příchozím směru 59,89 Mbit/s a v odchozím 44,03 Mbit/s.



Obrázek 4.13: Statistika přenosu zachycená na straně serveru při maximálně 1 000 transakcí

Adresář s výsledky kromě jiných obsahuje i data týkající se nižších vrstev ISO/OSI modelu, zde je zobrazen počet odeslaných/přijatých paketů/bitů. A to jak pro jednotlivé aplikační protokoly, tak součet všech odeslaných/přijatých paketů/bitů. V tabulce 4.1 lze vidět odchycený počet odeslaných/přijatých paketů. Z těchto dat lze jednoduše vypočítat ztrátovost paketů vzorcem:

$$Ztrátovost\ paketů = \frac{(Odeslané\ pakety - Přijaté\ pakety)}{Odeslané\ pakety} * 100\ [\%]$$

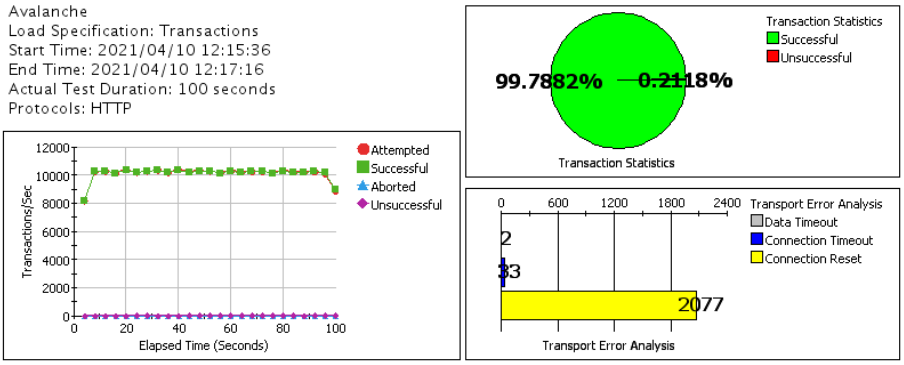
V odchozím směru je ztráta paketů 0,9 % a v příchozím 1,38 %. V příchozím směru je ztrátovost vyšší, neboť jsme generovali více paketů a strana serveru pouze odpovídala na naše dotazy. Ztrátovost je v tomto případě téměř zanedbatelná.

Tabulka 4.1: Přijaté a odeslané pakety při maximálně 1 000 transakcí (zařízení Spirent jako Klient i Server)

	Odeslané pakety	Přijaté pakety
Server	3081296	7019776
Klient	7118011	3053444

4.2.3 Testování při maximálně 1 000 transakcí na serveru Nginx

Po prvním měření jsem Spirent server nahradil serverem Nginx. Zde jsem použil stejnou zátěž, v tabulkách lze vidět, že počet neúspěšných transakcí je o něco vyšší než v prvním měření. Taktéž je vyšší zpoždění jednotlivých interakcí.



Test Results Summary	Transactions			Time (ms)					TCP Connections	
		Total	Rate Per Second		Page Response	URL Response	To TCP SYN/ACK	To First Data Byte	Est. Server Response		Total
	Attempted	1014009	10140	Minimum	0.0	0.0	0.312	0.608	0.0	Attempted	1014002
	Successful	1011861	10118	Maximum	10187.0	10187.0	7022.935	10187.761	10183.67	Established	1013940
	Unsuccessful	2148	21	Average	16.218	16.217	46.436	16.216	13.521		
	Aborted	0	0								

User Profile Definitions	Test							
	Profile	Percentage	Think Time (seconds)	Aborted	Protocol	Persistent Connection	Transactions Per Connection	Connections Per Server
	Default_0	100.000%	5.0	0 % after 0 seconds	HTTP 1.1	Enabled	10	4

Transaction Summary	Test	Count	Transactions (Sub-Commands included)								Response Time (us)		
	Profile	URL	Average Successful Per Second	Attempted	Successful	Unsuccessful	Aborted	Percent Successful	Percent Unsuccessful	Percent Aborted	Minimum	Maximum	Average
	Default_0	2	10118	1014009	1011861	2148	0	99.78	0.21	0.0	610.0	1.0187763E7	16218.154
	Totals	2		1014009	1011861	2148	0	99.78	0.21	0.0			

Obrázek 4.14: Souhrnná statistika testování při maximálně 1 000 transakcí (zařízení Spirent pouze jako Klient)

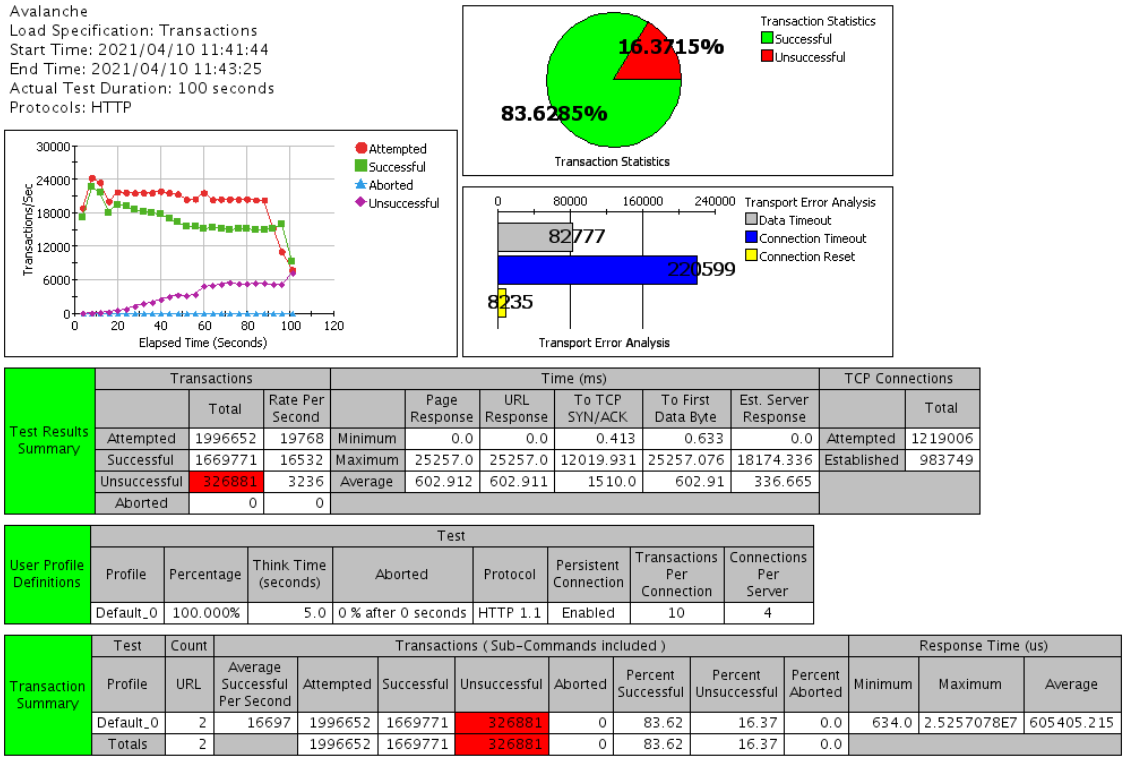
Při maximálně 1 000 transakcí s použitím serveru Nginx je ztrátovost paketů v odchozím směru 1,1 % a v příchozím 1,7 %. Hodnoty jsou opět o něco horší než u prvního měření, jelikož je veškeré nastavení totožné, tak je zřejmé, že zhoršení je způsobeno Nginx serverem, který je výkonnostně horší než Spirent server.

Tabulka 4.2: Přijaté a odeslané pakety při maximálně 1 000 transakcí (zařízení Spirent pouze jako Klient)

	Odeslané pakety	Přijaté pakety
Server	5143600	5084947
Klient	5174272	5086645

4.2.4 Testování při maximálně 100 000 transakcí na serveru zařízení Spirent

Ve třetím měření jsem generoval zátěž se stejným tvarem jako u prvních dvou měření s tím rozdílem, že maximální počet generovaných transakcí byl 100 000. Kromě zvýšeného provozu lze v tabulkách zahlédnout mnohem vyšší počet neúspěšných transakcí a zvýšené zpoždění prováděných interakcí z jednotek až desítek ms na stovky ms.



Test Results Summary	Transactions			Time (ms)						TCP Connections	
		Total	Rate Per Second		Page Response	URL Response	To TCP SYN/ACK	To First Data Byte	Est. Server Response		Total
	Attempted	1996652	19768	Minimum	0.0	0.0	0.413	0.633	0.0	Attempted	1219006
	Successful	1669771	16532	Maximum	25257.0	25257.0	12019.931	25257.076	18174.336	Established	983749
	Unsuccessful	326881	3236	Average	602.912	602.911	1510.0	602.91	336.665		
	Aborted	0	0								

User Profile Definitions	Test							
	Profile	Percentage	Think Time (seconds)	Aborted	Protocol	Persistent Connection	Transactions Per Connection	Connections Per Server
	Default_0	100.000%	5.0	0 % after 0 seconds	HTTP 1.1	Enabled	10	4

Transaction Summary	Test	Count	Transactions (Sub-Commands included)							Response Time (us)		
	Profile	URL	Average Successful Per Second	Attempted	Successful	Unsuccessful	Aborted	Percent Successful	Percent Unsuccessful	Percent Aborted	Minimum	Maximum	Average
	Default_0	2	16697	1996652	1669771	326881	0	83.62	16.37	0.0	634.0	2.5257078E7	605405.215
	Totals	2		1996652	1669771	326881	0	83.62	16.37	0.0			

Obrázek 4.15: Souhrnná statistika testování při maximálně 100 000 transakcí (zařízení Spirent jako Klient i Server)

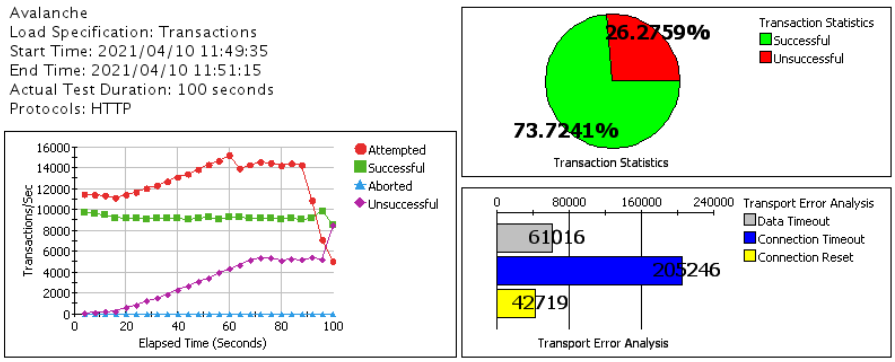
Ze statistik na straně serveru víme, že propustnost sítě byla v příchozím směru 50,96 Mbit/s a v odchozím 55,92 Mbit/s. Při maximálně 100 000 transakcí je ztrátovost paketů v odchozím směru 9,29 % a v příchozím směru 36,77 %. Tato ztrátovost absolutně nepřijatelná, na souhrnné statistice lze vidět, že i jednotlivé zpoždění jsou daleko vyšší než u měření s maximálně 1 000 transakcí.

Tabulka 4.3: Přijaté a odeslané pakety při maximálně 100 000 transakcí (zařízení Spirent jako Klient i Server)

	Odeslané pakety	Přijaté pakety
Server	4247204	6070978
Klient	9602406	3852465

4.2.5 Testování při maximálně 100 000 transakcí na serveru Nginx

Podobně jako u měření na serveru Spirent došlo k nárůstu zpoždění, taktéž se zde projevuje trend z měření při maximálně 1 000 transakcí, kde byla úspěšnost provedených transakcí nižší na Nginx serveru.



Test Results Summary	Transactions			Time (ms)						TCP Connections	
		Total	Rate Per Second		Page Response	URL Response	To TCP SYN/ACK	To First Data Byte	Est. Server Response		Total
	Attempted	1253219	12532	Minimum	0.0	0.0	0.315	0.611	0.0	Attempted	1217095
	Successful	923924	9239	Maximum	56105.0	56105.0	13340.352	56105.43	54099.754	Established	996364
	Unsuccessful	329295	3292	Average	1515.5	1515.499	1477.627	1515.497	946.053		
	Aborted	0	0								

User Profile Definitions	Test							
	Profile	Percentage	Think Time (seconds)	Aborted	Protocol	Persistent Connection	Transactions Per Connection	Connections Per Server
	Default_0	100.000%	5.0	0 % after 0 seconds	HTTP 1.1	Enabled	10	4

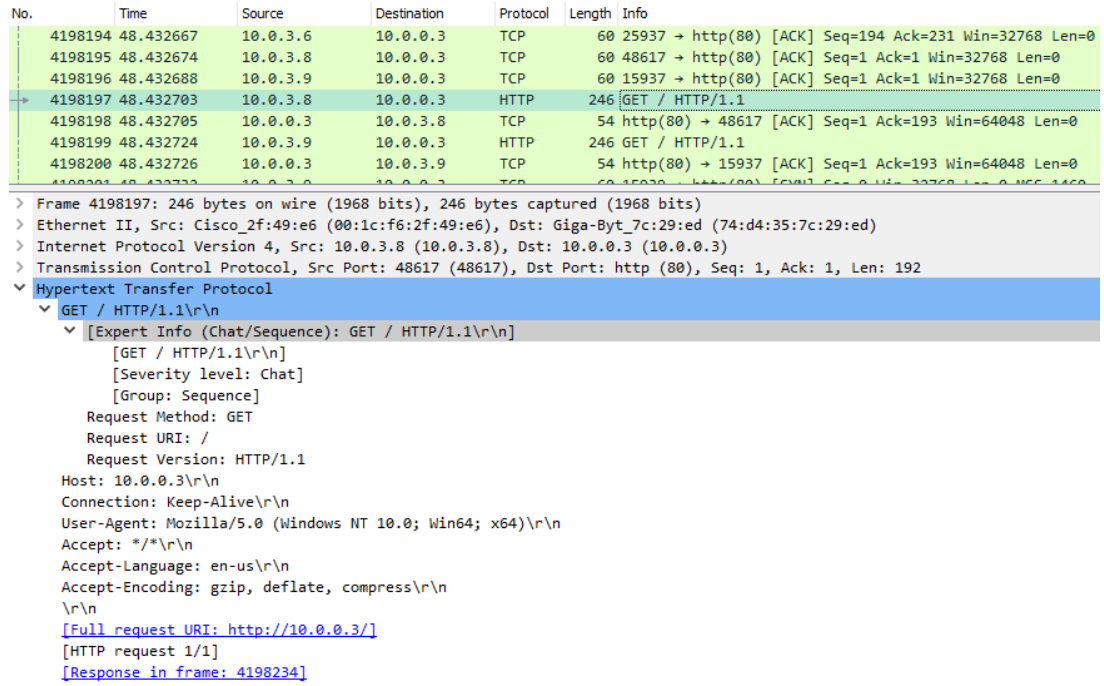
Transaction Summary	Test	Count	Transactions (Sub-Commands included)							Response Time (us)		
	Profile	URL	Average Successful Per Second	Attempted	Successful	Unsuccessful	Aborted	Percent Successful	Percent Unsuccessful	Percent Aborted	Minimum	Maximum	Average
	Default_0	2	9239	1253219	923924	329295	0	73.72	26.27	0.0	612.0	5.610543E7	1516380.095
	Totals	2		1253219	923924	329295	0	73.72	26.27	0.0			

Obrázek 4.16: Souhrnná statistika testování při maximálně 100 000 transakcí (zařízení Spirent pouze jako Klient)

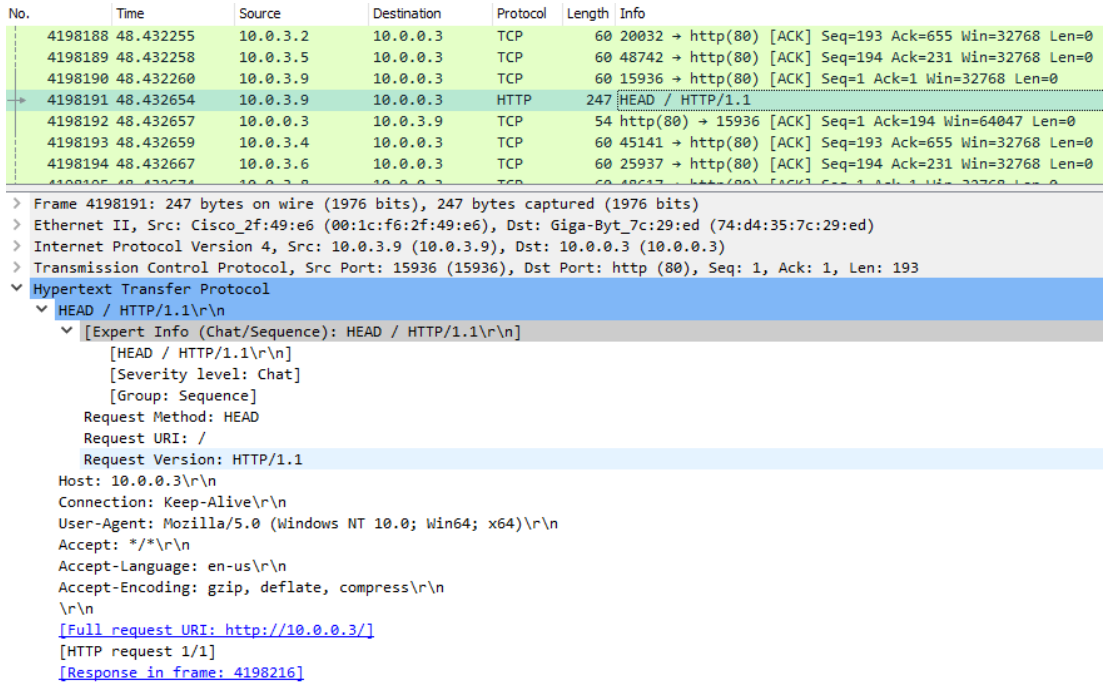
U měření na Nginx serveru při maximálně 100 000 transakcí je ztrátovost paketů v odchozím směru 27,62 % a v příchozím 37,23 %. Podobně jako u měření při maximálně 1 000 transakcí jsou zde opět horší hodnoty než při měření na zařízení Spirent.

Tabulka 4.4: Přijaté a odeslané pakety při maximálně 100 000 transakcí (zařízení Spirent pouze jako Klient)

	Odeslané pakety	Přijaté pakety
Server	6116032	5046281
Klient	8040378	4426656



Obrázek 4.17: Metoda GET

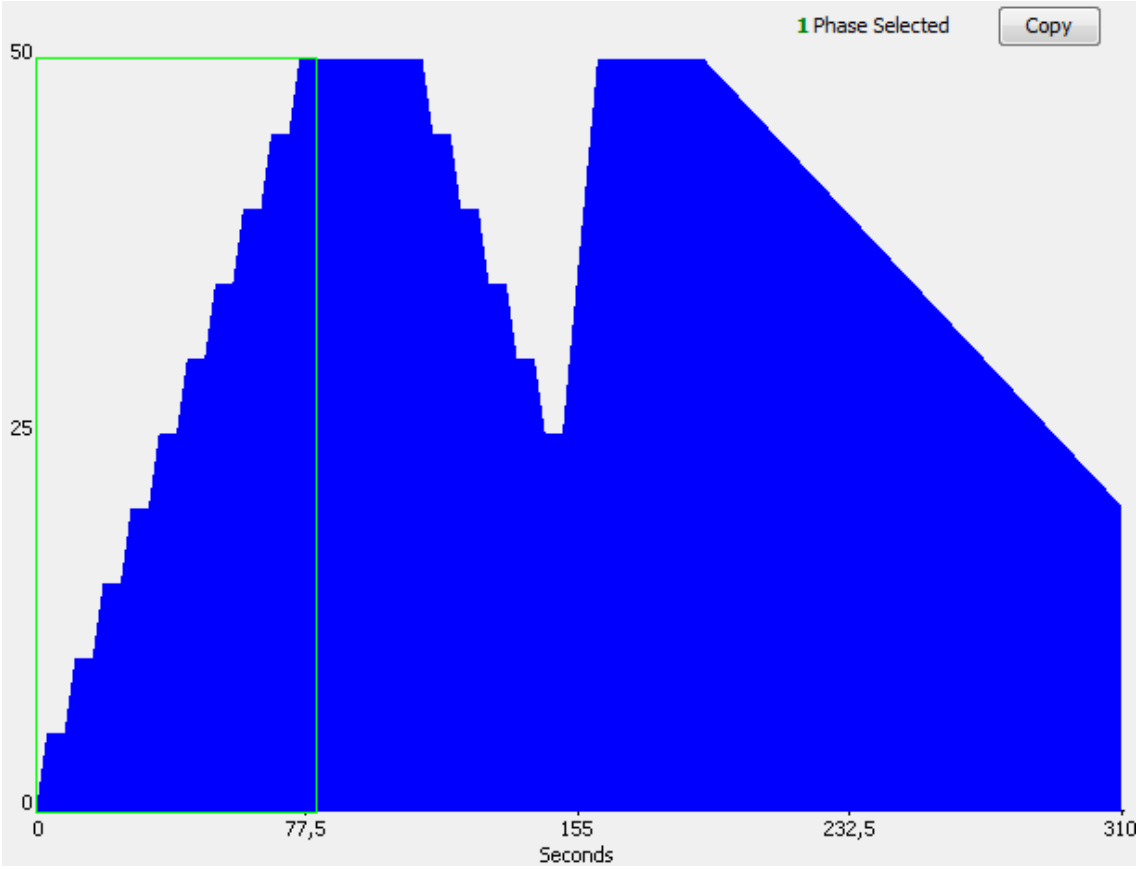


Obrázek 4.18: Metoda HEAD

Na obrázcích 4.17 a 4.18 je odchycena komunikace pomocí nástroje TCPDUMP mezi stranou klienta zařízení Spirent a Nginx serverem. Na prvním obrázku lze vidět žádost o zaslání požadované stránky včetně následné TCP komunikace, na druhém obrázku je odchycen druhou žádost HEAD. Po otevření detailů HTTP protokolu lze v hlavičce vidět například typ žádosti, verze HTTP, typ spojení či prohlížeče.

4.3 Testování protokolu FTP

Měření probíhalo na topologii z podkapitoly 4.2 (obrázek 4.10). Provoz byl generován z IP adres 10.0.3.2-10.0.3.9. IP adresa serveru je opět 10.0.0.3. Zařízení Spirent bylo použito jak ke generování provozu, tak k simulaci FTP serveru.



Obrázek 4.19: Tvar zátěže - 50 spojení

U FTP byly provedeny 2 měření, první s maximálně 50 spojeními a druhé s maximálně 5 000 spojeními. Jelikož je generování zátěže definované pomocí počtu transakcí umožněno pouze protokolům HTTP a HTTPS, tak jsem zvolil alternativní možnost a definoval zátěž pomocí počtu spojení. Princip generování je podobný jako u transakcí s tím rozdílem, že během jednoho spojení může být provedeno několik transakcí. Definovat zátěž pomocí počtu spojení můžeme pouze u protokolů, které pracují na transportní vrstvě pod záštitou protokolu TCP. Protokoly, které pracují na transportní vrstvě pod protokolem UDP nenavazují spojení a tudíž, je nemožné takto definovat zátěž.

4.3.1 Nastavení zařízení Spirent

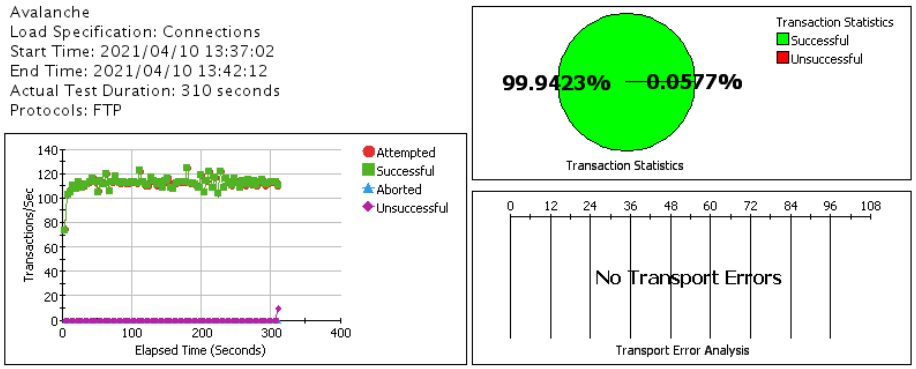
Na straně klienta byla generována metoda GET, kterou žádáme server o zaslání dat. Můžeme buďto žádat o abstraktní data s určitou velikostí nebo konkrétní soubory. Konkrétně jsem žádal server o zasílání souborů o velikosti 100 kbit, zároveň jsem přidal autentizaci a nastavil mód přenosu jako binární. Finální syntaxe vypadala následovně:

```
1 ftp://10.0.0.3/100k <USER=student PASSWD=student>
<MODE=BINARY>
```

Na straně serveru jsem nastavil protokol na FTP a vypnul ignorování ARP/NDP žádostí. V záložce Authentications jsem přidal uživatele, kterým se budu připojovat k FTP serveru. Zbylé nastavení jak na straně serveru, tak na straně klienta zůstalo výchozí.

4.3.2 Testování při maximálně 50 spojeních

Na obrázku 4.20 můžeme vidět souhrnnou statistiku měření FTP protokolu při maximálně 50 spojeních, avšak oproti protokolu HTTP zde nejsou všechny důležité informace. V první tabulce můžeme vidět například počet úspěšných/neúspěšných transakcí, minimální, maximální a průměrné zpoždění odpovědi URL či odpovědi serveru s navázaným spojením, dále je zde celkový počet pokusů o spojení a celkový počet navázaných spojení. Parametry Rate Per Second (Transactions), Page Response a To First Data Byte (Time) neukazují korektní informace, neboť se nejedná o protokol HTTP/HTTPS. V druhé tabulce je nastavení uživatele, ten ve výchozím nastavení úmyslně nezahazuje žádné pakety a má okamžitou odezvu, protokol byl nastaven na FTP. Ve třetí tabulce jsou detailní parametry transakcí, tak jako v měření u HTTP.



Test Results Summary	Transactions			Time (ms)						TCP Connections	
		Total	Rate Per Second		Page Response	URL Response	To TCP SYN/ACK	To First Data Byte	Est. Server Response		Total
	Attempted	34692	-	Minimum	1.271310336E9	18.0	0.408	0.0	0.0	Attempted	69384
	Successful	34672	-	Maximum	0.0	448.0	42.672	0.0	19.495	Established	69384
	Unsuccessful	20	-	Average		316.787	2.453	0.0	1.681		
	Aborted	0	-								

User Profile Definitions	Test							
	Profile	Percentage	Think Time (seconds)	Aborted	Protocol	Persistent Connection	Transactions Per Connection	Connections Per Server
	Default_0	100.000%	0.0	0 % after 0 seconds	FTP	Enabled	10	4

Transaction Summary	Test	Count	Transactions (Sub-Commands included)							Response Time (us)		
	Profile	URL	Average Successful Per Second	Attempted	Successful	Unsuccessful	Aborted	Percent Successful	Percent Unsuccessful	Percent Aborted	Minimum	Maximum	Average
	Default_0	1	111	34692	34672	20	0	99.94	0.05	0.0	18549.0	448070.0	316786.745
	Totals	1		34692	34672	20	0	99.94	0.05	0.0			

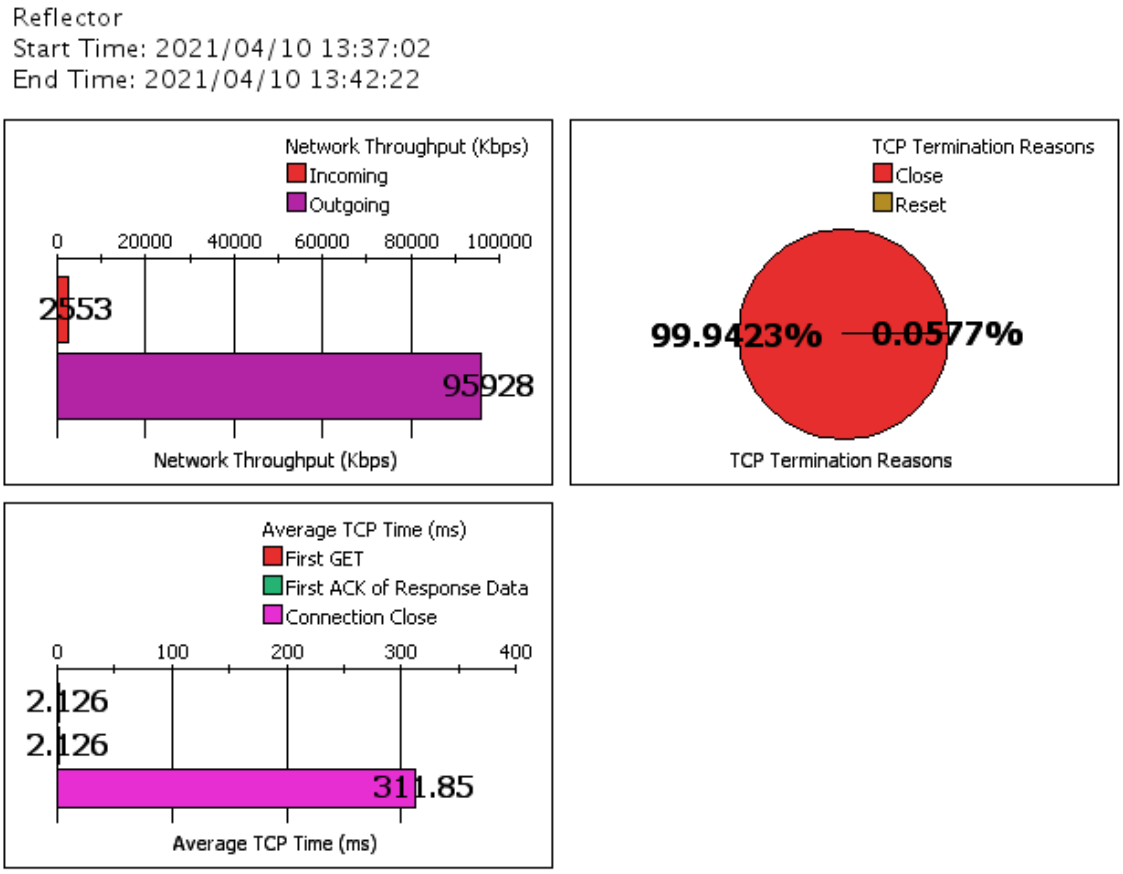
Obrázek 4.20: Souhrnná statistika při maximálně 50 spojeních

Pokud chceme detailnější FTP výsledky, tak ve výsledcích měření musíme otevřít záložku týkající se FTP (FTP Summary). Kromě parametrů uvedených v tabulce 4.5 nalezneme v FTP Summary například i počet jednotlivých FTP příkazů.

Tabulka 4.5: FTP statistika při maximálně 50 spojeních

Celkový počet pokusů o navázání FTP relace	34692
Celkový počet úspěšně navázaných řídících FTP relací	34672
Celkový počet úspěšně navázaných datových FTP relací	34672
Celkový počet neúspěšně navázaných řídících FTP relací	20
Celkový počet zahozených řídících FTP relací	0
Celkový počet přenesených dat protokolem FTP	3468,112 MB
Celkový počet FTP příkazů	208132
Průměrná/Maximální/Minimální rychlost FTP přenosu	111,845/122/73,75 souborů/s
Průměrná doba řídícího spojení FTP	316,65 ms
Průměrná doba FTP přihlašování	1,6 ms
Průměrná doba datového spojení FTP	295,38 ms

V 99,94 % došlo ke korektnímu ukončení spojení, ve zbytku případů došlo k resetování spojení. Propustnost sítě je v příchozím směru 2,55 Mbit/s a v odchozím 95,93 Mbit/s.



Obrázek 4.21: Statistika zachycená na straně serveru při maximálně 50 spojeních

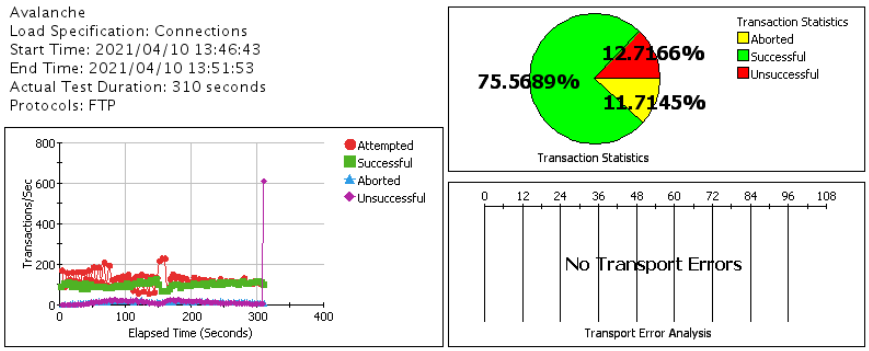
V odchozím směru je ztrátovost paketů 0,012 % a v příchozím 0 %. Takto dobrá ztrátovost je způsobena nízkou generovanou zátěží.

Tabulka 4.6: Přijaté a odeslané pakety při maximálně 50 spojeních

	Odeslané pakety	Přijaté pakety
Server	3087007	1644977
Klient	1644977	3086644

4.3.3 Testování při maximálně 5 000 spojeních

V důsledku velkého počtu generovaných spojení docházelo k nucenému zahazování transakcí s čímž se pojí i zvýšené procento neúspěšných transakcí. Zpoždění příznaků SYN/ACK a komunikace při navázaném spojení je zde oproti předchozímu měření více než dvěstěnásobné.



Test Results Summary	Transactions			Time (ms)						TCP Connections	
		Total	Rate Per Second		Page Response	URL Response	To TCP SYN/ACK	To First Data Byte	Est. Server Response		Total
	Attempted	39020	-	Minimum	1.271310336E9	20.0	0.425	0.0	0.0	Attempted	76369
	Successful	29487	-	Maximum	0.0	211425.0	12113.143	0.0	48989.383	Established	76294
	Unsuccessful	4962	-	Average		16359.297	489.471	0.0	342.053		
	Aborted	4571	-								

User Profile Definitions	Test							
	Profile	Percentage	Think Time (seconds)	Aborted	Protocol	Persistent Connection	Transactions Per Connection	Connections Per Server
	Default_0	100.000%	0.0	0 % after 0 seconds	FTP	Enabled	10	4

Transaction Summary	Test	Count	Transactions (Sub-Commands included)								Response Time (us)		
	Profile	URL	Average Successful Per Second	Attempted	Successful	Unsuccessful	Aborted	Percent Successful	Percent Unsuccessful	Percent Aborted	Minimum	Maximum	Average
	Default_0	1	95	39020	29487	4962	4571	75.56	12.71	11.71	20198.0	2.11425529E8	1.6359297329E7
	Totals	1		39020	29487	4962	4571	75.56	12.71	11.71			

Obrázek 4.22: Souhrnná statistika při maximálně 5 000 spojeních

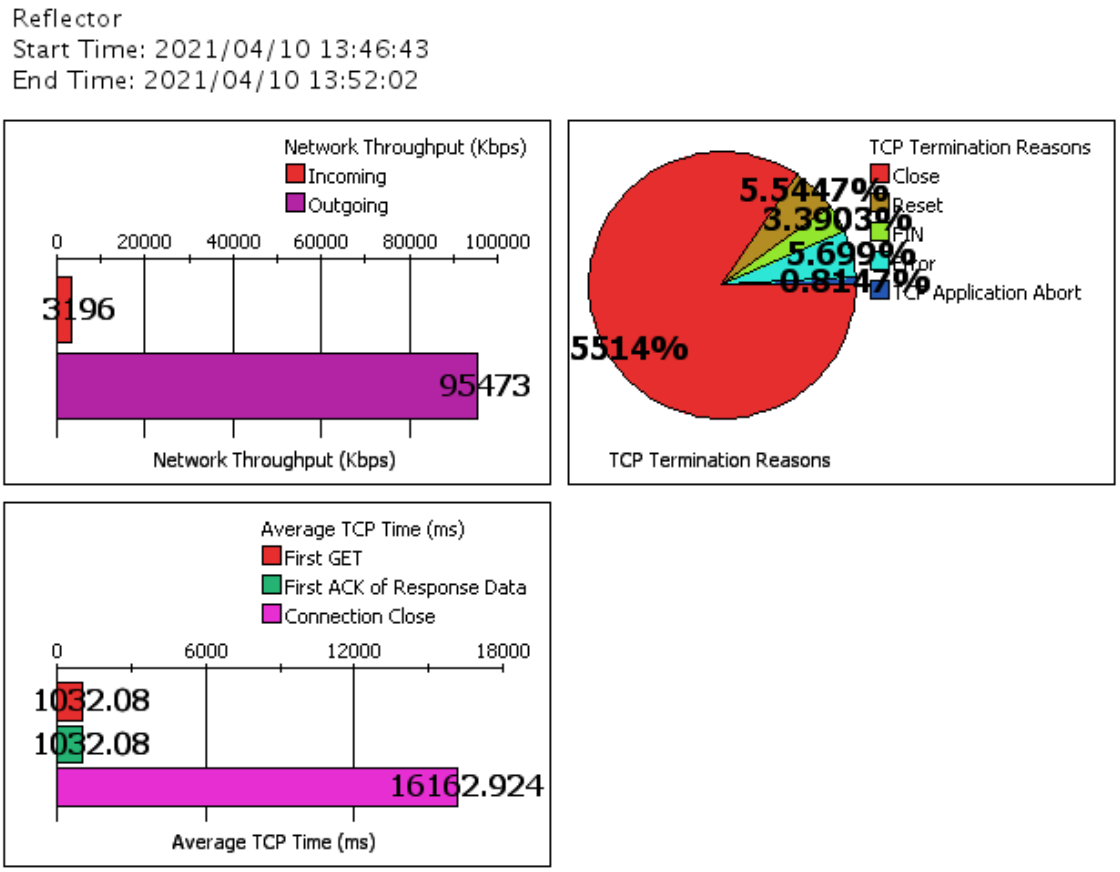
Ve druhém měření byl zvýšen počet pokusů o navázání FTP relace přibližně o 5000, počet úspěšně navázaných řídících a datových relací je však výrazně nižší než v měření prvním. Taktéž došlo k nárůstu příkazů FTP. Rychlosti FTP přenosu jsou srovnatelné s prvním měřením. Průměrná doba jednotlivých spojení a přihlašování je několikanásobně vyšší než v prvním měření.

Tabulka 4.7: FTP statistika při maximálně 5 000 spojeních

Celkový počet pokusů o navázání FTP relace	39020
Celkový počet úspěšně navázaných řídících FTP relací	29487
Celkový počet úspěšně navázaných datových FTP relací	33534

Celkový počet neúspěšně navázaných řídících FTP relací	4962
Celkový počet zahozených řídících FTP relací	4571
Celkový počet přenesených dat protokolem FTP	3518,751 MB
Celkový počet FTP příkazů	223858
Průměrná/Maximální/Minimální rychlost FTP přenosu	108,17/129,5/74,52 souborů/s
Průměrná doba řídícího spojení FTP	16837 ms
Průměrná doba FTP přihlašování	987,1 ms
Průměrná doba datového spojení FTP	12038 ms

V 84,55 % došlo ke korektnímu ukončení spojení. Jelikož zátěž byla vyšší, než byl server schopen pojmout docházelo i k jiným způsobům ukončení spojení. V 5,55 % bylo spojení resetováno příznakem RST, v 3,39 % bylo spojení ukončeno na základě příznaku FIN, 5,67 % tvořila chyba a v 0,82 % bylo TCP spojení zahozeno. Propustnost sítě byla v příchozím směru 3,2 Mbit/s a v odchozím 95,47 Mbit/s.



Obrázek 4.23: Statistika zachycená na straně serveru při maximálně 5 000 spojeních

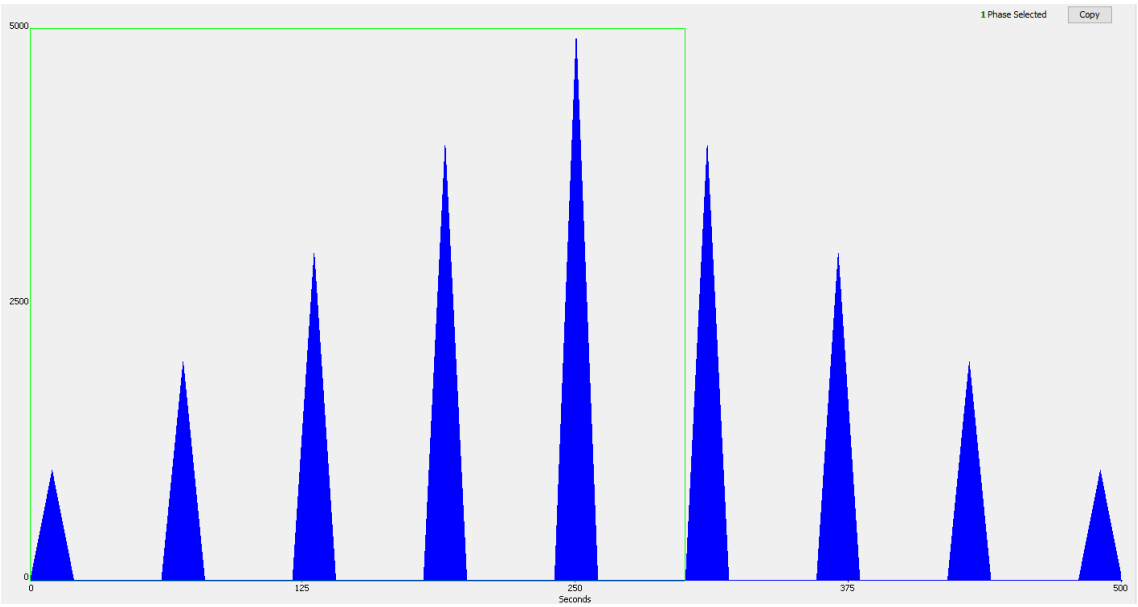
Ztrátovost paketů v odchozím směru činí 9,58 % a v příchozím směru 0,4 %. Ztrátovost paketů v odchozím směru je v normě, avšak v příchozím směru je ztrátovost příliš velká.

Tabulka 4.8: Přijaté a odeslané pakety při maximálně 5 000 spojeních

	Odeslané pakety	Přijaté pakety
Server	3529296	2099319
Klient	2107741	3191264

4.4 Testování protokolu SMTP

K testování byla opět využita topologie z podkapitoly 4.2 (obrázek 4.10). Pro stranu klienta byl využit rozsah IP adres 10.0.3.2-10.0.3.9. Server byl v síti pod IP adresou 10.0.0.3. Obě strany byly opět simulovány zařízením Spirent.



Obrázek 4.24: Tvar zátěže - 5 000 simulovaných uživatelů

Byly provedeny 2 měření podobně jako u testování protokolu FTP. Tvar zátěže byl zvolen, tak aby bylo demonstrováno co nejvíce možných vzorů (patternů). Pro demonstraci možností generátoru byla zátěž definována počtem simulovaných uživatelů. Jednotliví simulovaní (virtuální) uživatelé vykonávají příkazy ze záložky Actions. Principiálně funguje stejně jako definování pomocí transakcí a spojení. Hlavní rozdíl je, že nepřestává snižovat generovanou zátěž, pokud dojde k selhání serveru. Běžnější je však definování zátěže za jednotku času. Může se jednat například o počet transakcí/s, spojení/s nebo simulovaných uživatelů/s. Nemusí se jednat pouze o sekundy, můžou to být i hodiny. Zde dochází během každé definované jednotky času k nárůstu transakcí/spojení/simulovaných uživatelů na základě tvaru zátěže.

4.4.1 Nastavení zařízení Spirent

Jelikož je SMTP protokol sloužící k posílání e-mailů, tak generátor umožňuje pouze jediný příkaz. Základní syntaxe SMTP protokolu musí obsahovat IP adresu SMTP serveru,

velikost samotné zprávy a e-mailovou adresu odesílatele a příjemce. V mém měření byly použity následující příkazy. Prvním příkazem posílám e-maily z test@test.cz na student@student.cz s velikostí zprávy 500 Bytů. Druhým příkazem posílám e-maily ze student@student.cz na test@test.cz s velikostí zprávy 500 Bytů.

```
smtp://10.0.0.5 FROM=<test@test.cz> TO=<student@student.cz>
DATA=<FIXED, 500>
```

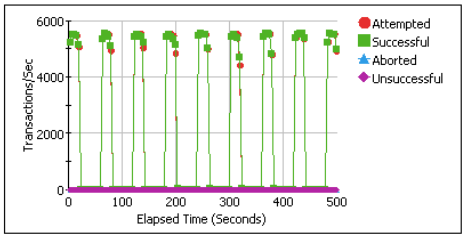
```
smtp://10.0.0.5 FROM=<student@student.cz> TO=<test@test.cz>
DATA=<FIXED, 500>
```

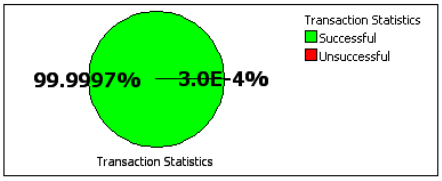
Na straně serveru jsem nastavil protokol na SMTP, deaktivoval vylepšenou verzi SMTP (ESMTP) a také jsem vypnul ignorování ARP/NDP žádostí. Zbylé nastavení na obou stranách bylo ponecháno výchozí.

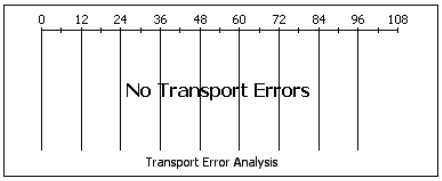
4.4.2 Testování při maximálně 5 000 simulovaných uživatelů

Podobně jako u testování protokolu FTP nám zde souhrnná statistika (obrázek 4.25) neřekne veškeré potřebné informace týkající se generovaného SMTP provozu.

Avalanche
Load Specification: SimUsers
Start Time: 2021/04/10 12:54:44
End Time: 2021/04/10 13:03:04
Actual Test Duration: 500 seconds
Protocols: SMTP







Test Results Summary	Transactions			Time (ms)						TCP Connections	
		Total	Rate Per Second		Page Response	URL Response	To TCP SYN/ACK	To First Data Byte	Est. Server Response		Total
	Attempted	968058	-	Minimum	1.271310336E9	4.0	0.411	0.0	0.0	Attempted	968058
	Successful	968055	-	Maximum	0.0	14110.0	6003.685	0.0	6001.45	Established	968056
	Unsuccessful	3	-	Average		47.624	7.759	0.0	4.24		
	Aborted	0	-								

User Profile Definitions	Test							
	Profile	Percentage	Think Time (seconds)	Aborted	Protocol	Persistent Connection	Transactions Per Connection	Connections Per Server
	Default_0	100.000%	0.0	0 % after 0 seconds	SMTP	Enabled	10	4

Transaction Summary	Test	Count	Transactions (Sub-Commands included)							Response Time (us)		
	Profile	URL	Average Successful Per Second	Attempted	Successful	Unsuccessful	Aborted	Percent Successful	Percent Unsuccessful	Percent Aborted	Minimum	Maximum	Average
	Default_0	2	1936	968058	968055	3	0	99.99	0.0	0.0	4285.0	1.4110193E7	47623.517
	Totals	2		968058	968055	3	0	99.99	0.0	0.0			

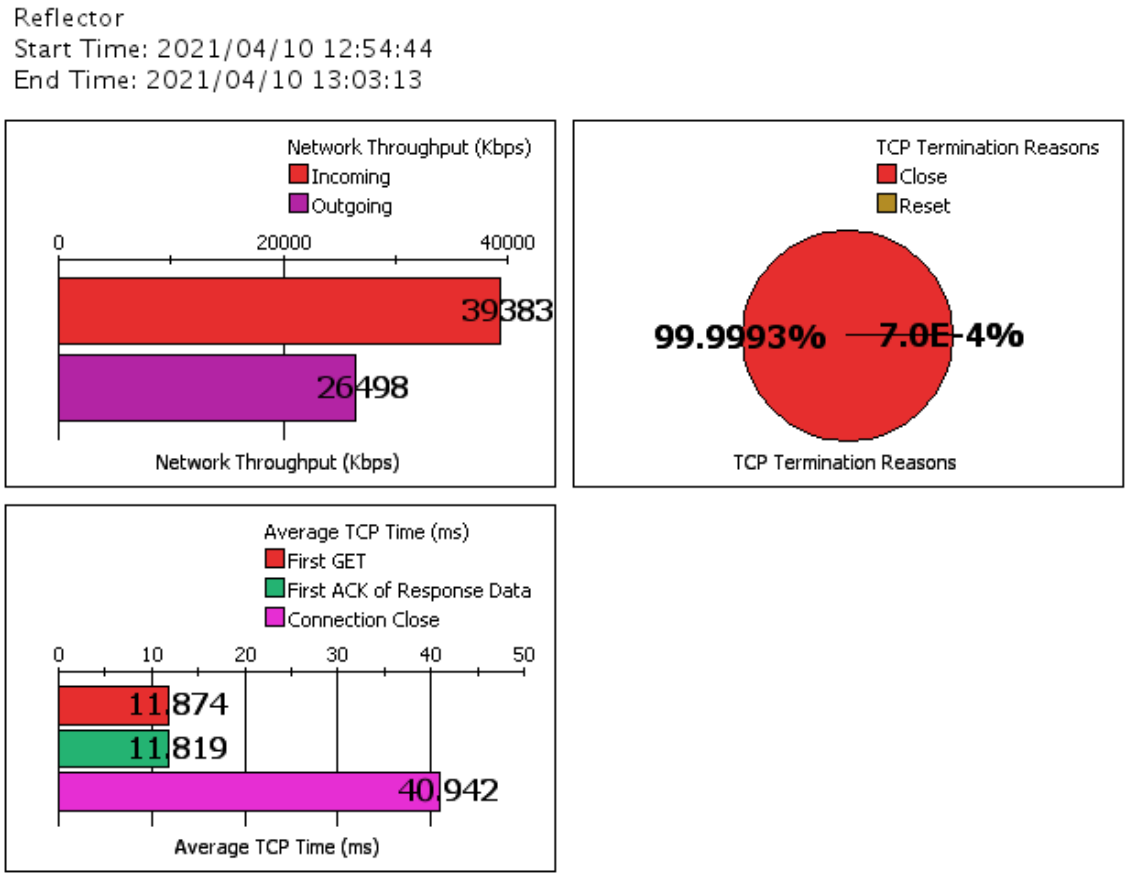
Obrázek 4.25: Souhrnná statistika při maximálně 5 000 simulovaných uživatelů

Pro zobrazení detailnějších informací týkajících se protokolu SMTP musíme otevřít záložku SMTP Summary. Veškeré data s nenulovými hodnotami jsem přepsal do tabulky 4.9. Protokol SMTP dokáže využívat několik druhů SMTP příkazů a speciálních kódů, při mém měření však byly využity pouze některé z nich.

Tabulka 4.9: SMTP statistika při 5 000 simulovaných uživatelů

Celkový počet pokusů o navázání SMTP relace	968058
Celkový počet úspěšně navázaných SMTP relací	968055
Celkový počet neúspěšně navázaných SMTP relací	3
Počet úspěšných/neúspěšných příkazů SMTP Greeting	968056/0
Počet úspěšných/neúspěšných příkazů SMTP Helo	968056/0
Počet úspěšných/neúspěšných příkazů SMTP Mail	968056/0
Počet úspěšných/neúspěšných příkazů SMTP Rcpt	968055/0
Počet úspěšných/neúspěšných příkazů SMTP Data	968055/0
Počet úspěšných/neúspěšných příkazů SMTP Message	968055/0
Počet úspěšných/neúspěšných příkazů SMTP Quit	968054/0
Počet použitých SMTP kódů 220 Service ready	968056
Počet použitých SMTP kódů 221 closing transmission	968054
Počet použitých SMTP kódů 250 action okay	3872222
Počet použitých SMTP kódů 354 Start mail input	968055

Korektně bylo spojení ukončeno až v 99,9993 % případů, v 0,0007 % případů bylo spojení resetováno. Propustnost v příchozím směru byla 39,38 Mbit/s a v odchozím 26,5 Mbit/s.



Obrázek 4.26: Statistika zachycená na straně serveru při maximálně 5 000 simulovaných uživatelů

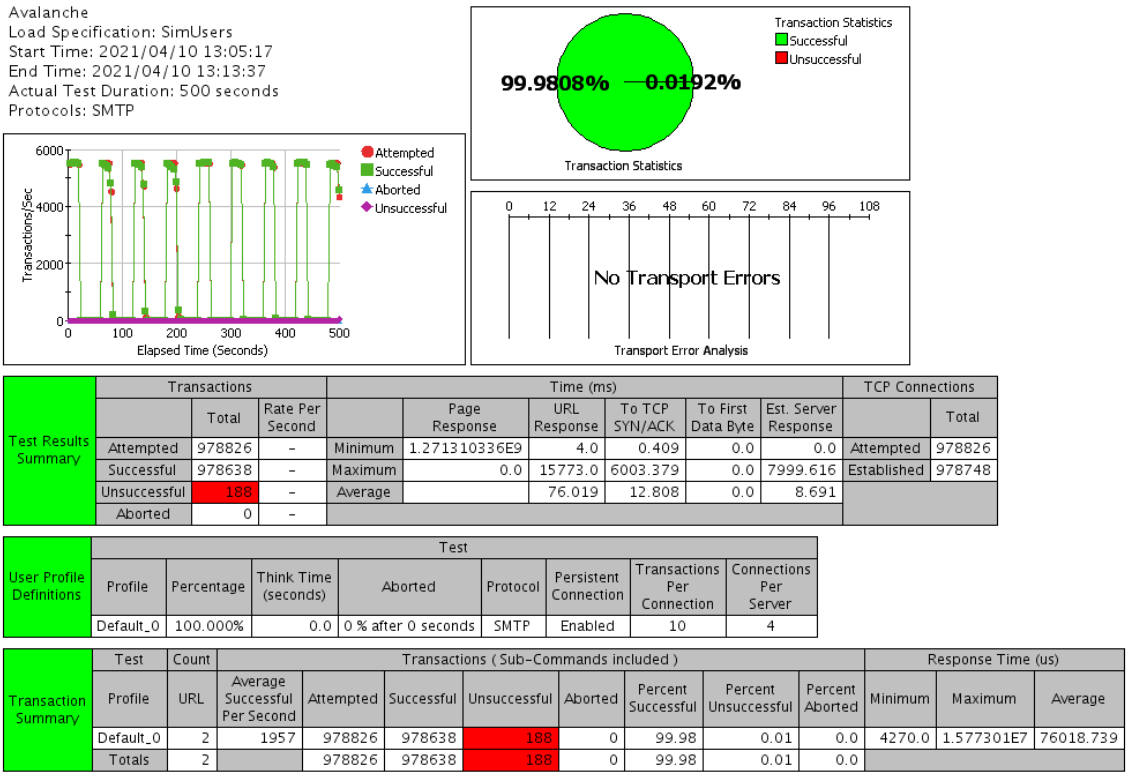
Ztrátovost paketů v odchozím směru je 0,1 % a v příchozím 0,18 %. V obou směrech přenosu je ztrátovost nízká, podobně jako u předchozích měření lze vidět korelaci mezi ztrátovostí paketů a propustností. Směr s vyšší propustností má zároveň vyšší ztrátovost paketů.

Tabulka 4.10: Přijaté a odeslané pakety při maximálně 5 000 simulovaných uživatelů

	Odeslané pakety	Přijaté pakety
Server	9708281	8730422
Klient	8746031	9698310

4.4.3 Testování při maximálně 50 000 simulovaných uživatelů

Ve druhém měření jsem použil desetkrát větší generovanou zátěž. Oproti předchozím testováním je zde chybovost stále nízká. Zvětšení zátěže nemělo velký vliv na zpoždění viz. první tabulka na obrázku 4.27. Z výsledků měření je zřejmé, že kromě velikosti zátěže má na úspěšnost vliv i tvar samotné zátěže a použitý protokol.



Test Results Summary	Transactions			Time (ms)						TCP Connections	
		Total	Rate Per Second		Page Response	URL Response	To TCP SYN/ACK	To First Data Byte	Est. Server Response		Total
	Attempted	978826	-	Minimum	1.271310336E9	4.0	0.409	0.0	0.0	Attempted	978826
	Successful	978638	-	Maximum	0.0	15773.0	6003.379	0.0	7999.616	Established	978748
	Unsuccessful	188	-	Average		76.019	12.808	0.0	8.691		
	Aborted	0	-								

User Profile Definitions	Test							
	Profile	Percentage	Think Time (seconds)	Aborted	Protocol	Persistent Connection	Transactions Per Connection	Connections Per Server
	Default_0	100.000%	0.0	0 % after 0 seconds	SMTP	Enabled	10	4

Transaction Summary	Test	Count	Transactions (Sub-Commands included)							Response Time (us)		
	Profile	URL	Average Successful Per Second	Attempted	Successful	Unsuccessful	Aborted	Percent Successful	Percent Unsuccessful	Percent Aborted	Minimum	Maximum	Average
	Default_0	2	1957	978826	978638	188	0	99.98	0.01	0.0	4270.0	1.577301E7	76018.739
	Totals	2		978826	978638	188	0	99.98	0.01	0.0			

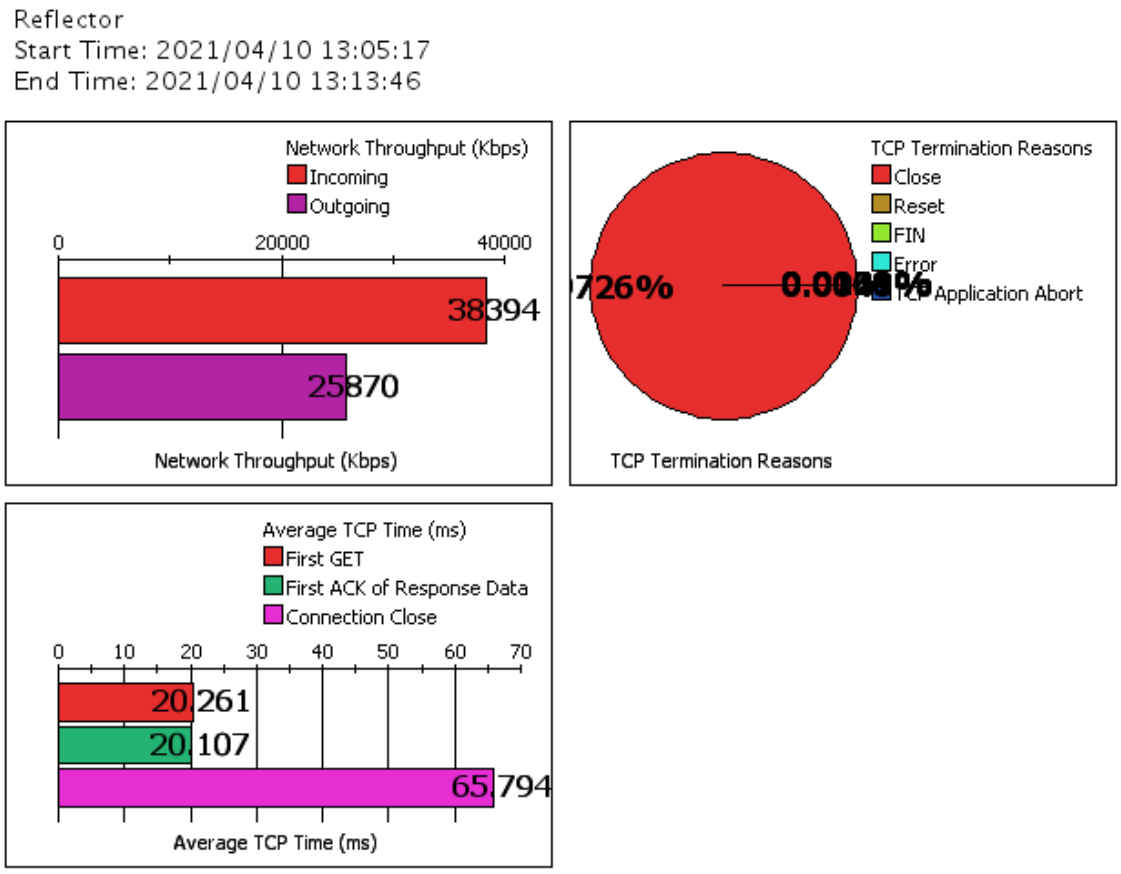
Obrázek 4.27: Souhrnná statistika při maximálně 50 000 simulovaných uživatelů

V rámci SMTP protokolu došlo k nárůstu počtu navázaných SMTP relací, příkazů a použitých kódu. Veškeré SMTP příkazy a kódy byly úspěšné tak jako v prvním SMTP měření.

Tabulka 4.11: SMTP statistika při 50 000 simulovaných uživatelů

Celkový počet pokusů o navázání SMTP relace	978826
Celkový počet úspěšně navázaných SMTP relací	978638
Celkový počet neúspěšně navázaných SMTP relací	188
Počet úspěšných/neúspěšných příkazů SMTP Greeting	978710/0
Počet úspěšných/neúspěšných příkazů SMTP Helo	978698/0
Počet úspěšných/neúspěšných příkazů SMTP Mail	978684/0
Počet úspěšných/neúspěšných příkazů SMTP Rcpt	978666/0
Počet úspěšných/neúspěšných příkazů SMTP Data	978646/0
Počet úspěšných/neúspěšných příkazů SMTP Message	978638/0
Počet úspěšných/neúspěšných příkazů SMTP Quit	978621/0
Počet použitých SMTP kódů 220 Service ready	978710
Počet použitých SMTP kódů 221 closing transmission	978621
Počet použitých SMTP kódů 250 action okay	3914686
Počet použitých SMTP kódů 354 Start mail input	978646

Při zvýšené zátěži se snížil počet standardně ukončených spojení na 99,97 % ve zbylých případech bylo spojení ukončeno resetováním, chybou, zahozením nebo na základě příznaku FIN. Propustnost v příchozím směru je 38,39 Mbit/s a v odchozím 25,87 Mbit/s.



Obrázek 4.28: Statistika zachycená na straně serveru při maximálně 50 000 simulovaných uživatelů

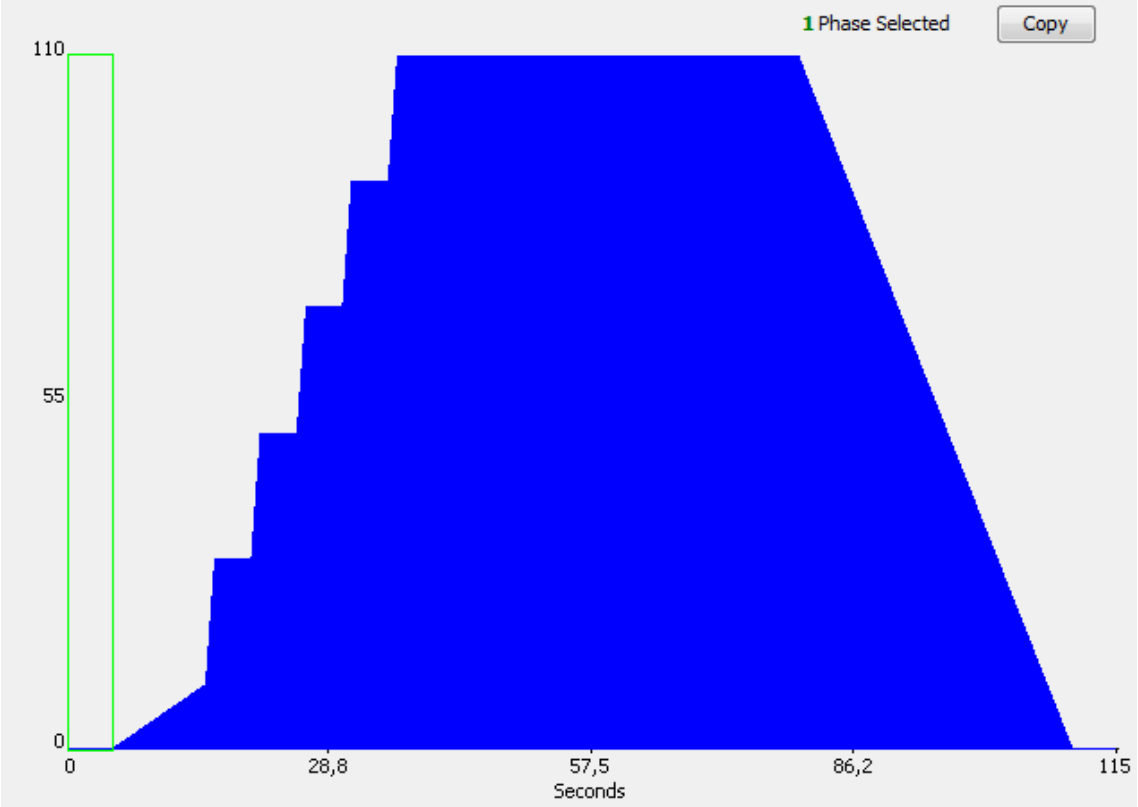
V odchozím směru je ztrátovost paketů 0,21 % a v příchozím 0,32 %. V obou směrech je ztrátovost dostatečně nízká a pro přenos dat nepředstavuje velký problém.

Tabulka 4.12: Přijaté a odeslané pakety při maximálně 50 000 simulovaných uživatelů

	Odeslané pakety	Přijaté pakety
Server	9840187	8841830
Klient	8871077	9819367

4.5 Testování několika aplikačních protokolů (HTTP, FTP, SMTP)

Pro poslední měření byla opět použita topologie z obrázku 4.10, strana klienta byla reprezentována IP adresami v rozsahu 10.0.3.2-10.0.3.9. Pro stranu serveru byly v tomto měření vyčleněny 3 IP adresy, pro HTTP server 10.0.0.3, pro FTP server 10.0.0.4 a pro SMTP server 10.0.0.5. Strana klienta i serveru byla realizována zařízením Spirent TestCenter C1.



Obrázek 4.29: Tvar zátěže - 110 spojení

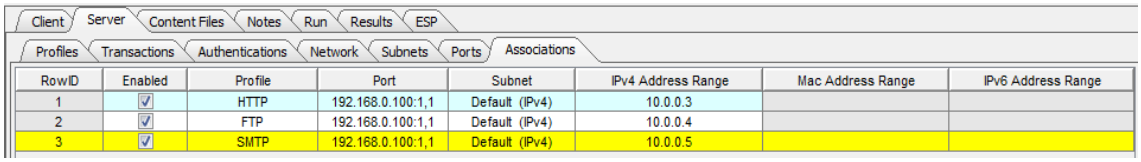
Na základě poznatků z předchozích měření byla generovaná zátěž zvolena tak, aby byla ztrátovost paketů co nejnižší. Zátěž byla definována pomocí počtu spojení.

4.5.1 Nastavení zařízení Spirent

Na straně klienta jsem generoval provoz tří různých protokolů. Veškeré použité příkazy již byly použity v předešlých měřeních, a tak není potřeba je znovu představovat. Ve finálním zápisu se změnily pouze IP adresy cílových serverů. Použitou syntaxi lze vidět na další straně.

```
1 get http://10.0.0.3/index.html

1 head http://10.0.0.3/index.html

1 ftp://10.0.0.4/100k <USER=student PASSWD=student>
<MODE=BINARY>

smtp://10.0.0.5 FROM=<test@test.cz> TO=<student@student.cz>
DATA=<FIXED, 500>

smtp://10.0.0.5 FROM=<student@student.cz> TO=<test@test.cz>
DATA=<FIXED, 500>
```

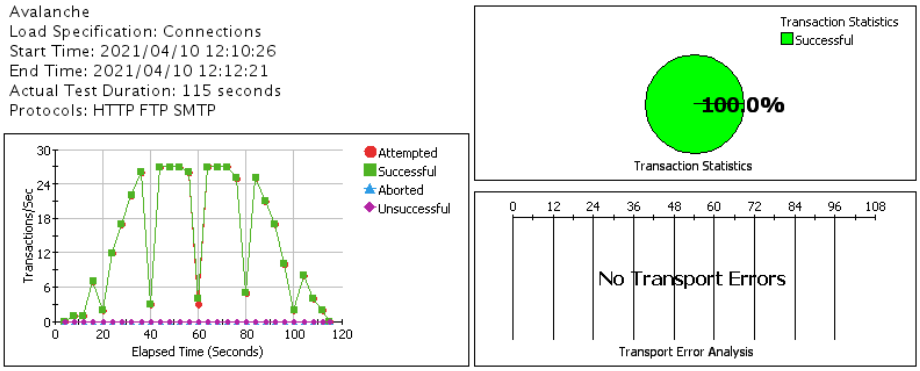
Na straně serveru jsem využil možnosti profilování. Vytvořil jsem tři různé profily, jeden pro protokol HTTP, druhý pro protokol FTP a třetí pro protokol SMTP. U každého z profilů jsem provedl totožné nastavení jako u dílčích testování. Nakonec jsem tyto profily spojil s IP adresami v záložce Associations, kterou lze vidět na obrázku 4.30. Zbylé nastavení bylo ponecháno jako výchozí.



Obrázek 4.30: Záložka Associations

4.5.2 Testování

Vlevo nahoře na obrázku 4.31 lze vidět, že opravdu generujeme 3 různé protokoly. Dále můžeme vidět 100 % úspěšnost transakcí a navázaných spojení. Zpoždění jednotlivých interakcí nepřesahuje jednotky milisekund.



Test Results Summary	Transactions			Time (ms)					TCP Connections	
		Total	Rate Per Second		Page Response	URL Response	To TCP SYN/ACK	To First Data Byte	Est. Server Response		Total
	Attempted	1605	13	Minimum	0.0	0.0	0.417	0.641	0.0	Attempted	1605
	Successful	1605	13	Maximum	3.0	2025.0	2000.44	3.032	1.219	Established	1605
	Unsuccessful	0	0	Average	0.689	8.386	2.235	0.687	0.239		
	Aborted	0	0								

User Profile Definitions	Test							
	Profile	Percentage	Think Time (seconds)	Aborted	Protocol	Persistent Connection	Transactions Per Connection	Connections Per Server
	Default_0	100.000%	5.0	0 % after 0 seconds	HTTP 1.1,, FTP,, SMTP	Enabled	10	4

Transaction Summary	Test	Count	Transactions (Sub-Commands included)							Response Time (us)		
	Profile	URL	Average Successful Per Second	Attempted	Successful	Unsuccessful	Aborted	Percent Successful	Percent Unsuccessful	Percent Aborted	Minimum	Maximum	Average
	Default_0	5	13	1605	1605	0	0	100.0	0.0	0.0	642.0	2025113.0	8382.297
	Totals	5		1605	1605	0	0	100.0	0.0	0.0			

Obrázek 4.31: Souhrnná statistika několika generovaných aplikačních protokolů (HTTP, FTP, SMTP)

Celkem bylo navázáno 1605 spojení z čehož bylo 322 FTP spojení, 639 SMTP spojení a zbylých 644 spojení proběhlo v rámci protokolu HTTP. Podle dat z obrázku 4.31 a tabulek 4.13, 4.14 můžeme říct, že přenos dat proběhl se 100 % úspěšností.

Tabulka 4.13: FTP statistika při maximálně 110 spojeních

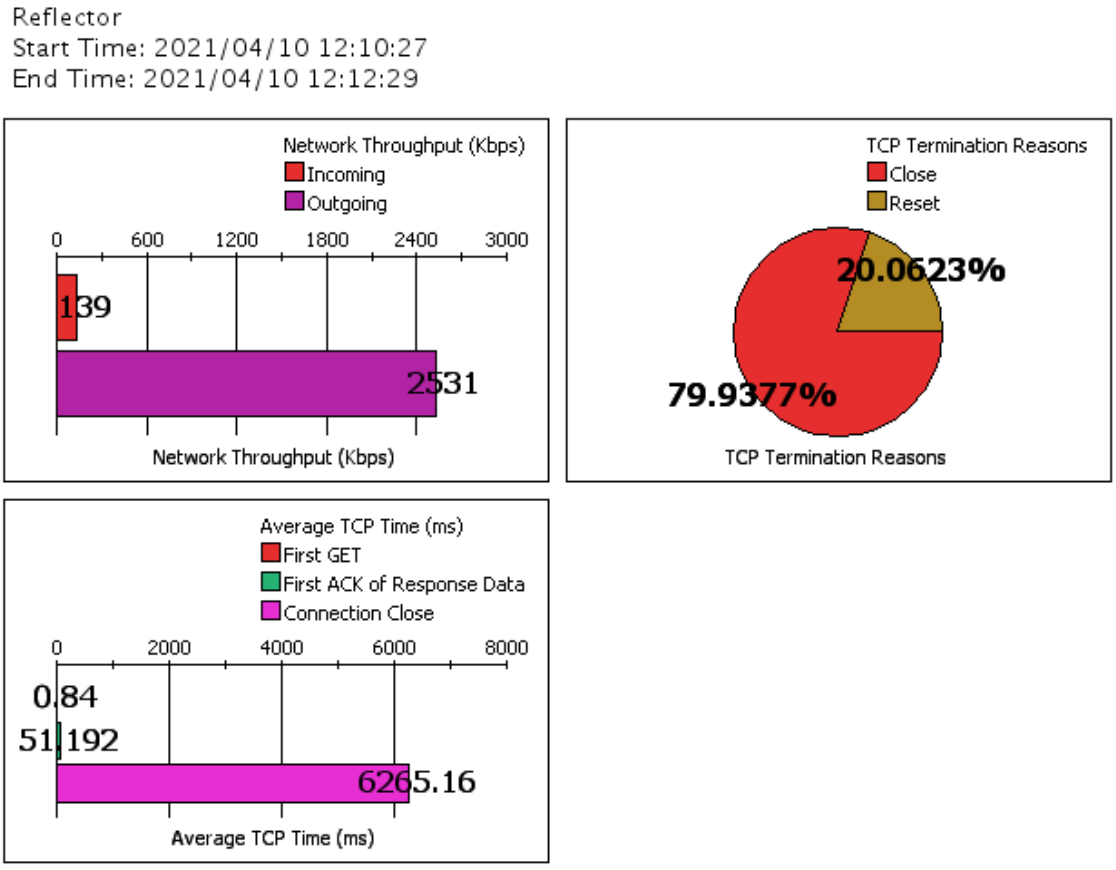
Celkový počet pokusů o navázání FTP relace	322
Celkový počet úspěšně navázaných řídících FTP relací	322
Celkový počet úspěšně navázaných datových FTP relací	332
Celkový počet neúspěšně navázaných řídících FTP relací	0
Celkový počet zahozených řídících FTP relací	0
Celkový počet přenesených dat protokolem FTP	32,2 MB
Celkový počet FTP příkazů	1932
Průměrná/Maximální/Minimální rychlost FTP přenosu	2,8/6,5/0 souborů/s
Průměrná doba řídícího spojení FTP	25,18 ms
Průměrná doba FTP přihlašování	0,92 ms
Průměrná doba datového spojení FTP	15,31 ms

Tabulka 4.14: SMTP statistika při maximálně 110 spojeních

Celkový počet pokusů o navázání SMTP relace	639
Celkový počet úspěšně navázaných SMTP relací	639
Celkový počet neúspěšně navázaných SMTP relací	0

Počet úspěšných/neúspěšných příkazů SMTP Greeting	639/0
Počet úspěšných/neúspěšných příkazů SMTP Helo	639/0
Počet úspěšných/neúspěšných příkazů SMTP Mail	639/0
Počet úspěšných/neúspěšných příkazů SMTP Rcpt	639/0
Počet úspěšných/neúspěšných příkazů SMTP Data	639/0
Počet úspěšných/neúspěšných příkazů SMTP Message	639/0
Počet úspěšných/neúspěšných příkazů SMTP Quit	639/0
Počet použitých SMTP kódů 220 Service ready	639
Počet použitých SMTP kódů 221 closing transmission	639
Počet použitých SMTP kódů 250 action okay	2556
Počet použitých SMTP kódů 354 Start mail input	639

V 79,94 % případů došlo ke korektnímu ukončení spojení. Ve 20,06 % bylo spojení resetováno, tento typ ukončení se nejvíce projevoval u protokolu HTTP. Propustnost v příchozím směru je 0,14 Mbit/s a v odchozím 2,53 Mbit/s.



Obrázek 4.32: Statistika zachycená na straně serveru při generování několika aplikačních protokolů (HTTP, FTP, SMTP)

Během celé komunikace došlo ke ztrátě pouhých 2 paketů, a to na straně klienta v odchozím směru. V odchozím směru je tudíž ztrátovost 0 % a v příchozím 0,009 %.

Tabulka 4.15: Přijaté a odeslané pakety při maximálně 110 spojeních - klient

	Odeslané pakety	Přijaté pakety
HTTP	2254	966
SMTP	5752	6390
FTP	14919	28658
Celkem	22925	36014

Tabulka 4.16: Přijaté a odeslané pakety při maximálně 110 spojeních - server

	Odeslané pakety	Přijaté pakety
HTTP	966	2254
SMTP	6390	5751
FTP	28658	14918
Celkem	36014	22923

5 Zhodnocení zařízení Spirent TestCenter C1 na základě praktické zkušenosti

Hlavní předností zařízení Spirent TestCenter C1 je velký počet funkcí, které umožňuje v oblasti generování a analýzy provozu. Zařízení disponuje několika programy, přičemž jeden z nich slouží ke generování obecného datového provozu a analýze parametrů přenosu, které jsou definovány v teorii o kvalitě služby. V dalším programu lze generovat provoz na úrovni aplikačních protokolů, tento program byl detailně popsán v této bakalářské práci. Mezi výhody tohoto programu patří možnost volby mezi generováním provozu, simulováním serveru a pouhým generováním provozu, možnost detailního generování jednotlivých zpráv v rámci daného aplikačního protokolu a jeho následná analýza na úrovni aplikační vrstvy. Program však není příliš vhodný k analýze parametrů kvality služby. Další nevýhodou je nemožnost odchytávat provoz v zapojení, kde zařízení Spirent slouží jako strana klienta i serveru, k těmto účelům je vhodnější první zmíněný program.

Spirent TestCenter C1, FETest Parascope GigE a iPerf umožňují analýzu provozu v rámci QoS. Testování jednotlivých aplikačních protokolů umožňují pouze Spirent TestCenter C1 a FETest Parascope GigE. Spirent však oproti FETestu dokáže detailně generovat datový provoz v rámci aplikačních protokolů, u FETestu můžeme pouze definovat použitý protokol, avšak nemůžeme upravovat generované zprávy. Testování pomocí FETestu je srovnatelné s nejjednodušším z dostupných testů (EZ test) programu Spirent TestCenter Layer 4-7 Application, kde pouze volíme typ aplikací a jejich procentuální podíl na generované zátěži. Zařízení Spirent nelze v rámci vyšších vrstev srovnávat s diagnostickým nástrojem iPerf, neboť iPerf umí diagnostikovat pouze protokoly nižších vrstev.

Závěr

Hlavním cílem této bakalářské práce bylo seznámit čtenáře se zařízením Spirent TestCenter C1 a jeho funkcemi v oblasti aplikačních protokolů. Mezi další cíle patří srovnání zařízení od Spirentu s jeho konkurencí, otestování alespoň 3 funkcí zařízení Spirent a sepsání návodů k použití zařízení Spirent TestCenter C1.

V praktické části bylo provedeno několik druhů měření aplikačních protokolů. Jako první jsem testoval protokol HTTP, kde jsem využil obou možností zapojení zařízení Spirent. V prvním měření jsem zapojil Spirent jako klienta a server. V druhém měření byl Spirent pouze jako klient a server (Nginx) byl vytvořen na počítači v laboratoři EB215 s operačním systémem Linux - Ubuntu. Další testované protokoly byly FTP a SMTP, kde jsem měnil velikost zátěže a sledoval jejich vliv na změnu přenosových parametrů. Hlavním smyslem těchto měření však bylo demonstrovat možnosti generátoru a následné analýzy datového provozu. V každém z měření byl záměrně zvolen jiný tvar zátěže pro demonstraci jednotlivých druhů vzorů (patternů). Dále bylo ukázáno, že zařízení může fungovat ve dvou režimech: klient a server nebo pouze klient. Při vytváření generované zátěže je možnost definovat zátěž několika způsoby, v mých měřeních jsem použil počet spojení, transakcí a simulovaných uživatelů. Taktéž jsem ukázal generování několika aplikačních protokolů v jednom testu, toho jsem dosáhl díky funkci, kterou zařízení umožňuje, jedná se o takzvané profilování a na základě vytvořených profilů jsem jednotlivým provozům přidělil jejich funkce. Detailní popis jednotlivých testů je popsán v přílohách včetně rozšířených možností jednotlivých protokolů.

V této práci jsem však zdaleka nevyužil všech funkcí zařízení Spirent. Kromě testování aplikačních protokolů umožňuje například generování různých útoku jako je Distributed Denial of Service. Na tuto práci by mohly navazovat práce týkající se výše zmíněného generování různých útoků a práce s druhým ze zmíněných programů, který je více zaměřen na parametry QoS a protokoly nižších vrstev.

Přínos této práce spočívá v teoretickém a praktickém srovnání zařízení Spirent TestCenter C1 vůči jiným generátorům/analyzátorům datového provozu a detailních návodech k použití zařízení Spirent TestCenter C1 se zaměřením na aplikační protokoly, které jsou v příloze.

Použitá literatura

[1] SZIGETI, Tim, Robert BARTON, Christina HATTINGH a Kenneth BRILEY. End-to-End QoS Network Design [online]. 2nd edition. Indianapolis, IN: Cisco Press, 2014 [cit. 2020-12-10]. ISBN 978-1-58714-369-4. Dostupné z: http://docshare04.docshare.tips/files/29270/292707884.pdf

[2] Kvalita služby (QoS) [online]. Brno: Fakulta informatiky Masarykovy univerzity, 2011 [cit. 2020-12-10]. Dostupné z: https://is.muni.cz/el/fi/jaro2011/PB156/um/lecture6.pdf

[3] KÁLLAY, Fedor a Peter PENIAK. Počítačové sítě a jejich aplikace. Praha: GRADA Publishing, 1999. ISBN 80-7169-407-X.

[4] KUROSE, James F., Keith W. ROSS a Jindřich JONÁK. Počítačové sítě. Brno: Computer Press, 2014. ISBN 978-80-251-3825-0.

[5] GRYGÁREK, Petr. Quality of Service (QoS) v IP sítích [online]. [cit. 2020-12-11]. Dostupné z: http://www.cs.vsb.cz/grygarek/SPS/lect/QoS/QoS.html

[6] GRYGÁREK, Petr. Podpora multimediálních aplikací v Internetu [online]. Ostrava [cit. 2020-12-11]. Dostupné z: http://www.cs.vsb.cz/grygarek/SPS/lect/multimedia-ucitele.pdf

[7] LAUTERBACH, Filip. Kvalita služby v sítích LAN. Ostrava, 2019. Bakalářská práce. VŠB - Technická univerzita Ostrava. Vedoucí práce Petr Machník.

[8] Types of Delay [online]. [cit. 2020-12-14]. Dostupné z: https://zahid-stanikzai.com/types-of-delay/

[9] CHRISTENSSON, Per. Bandwidth Definition. TechTerms [online]. 2012 [cit. 2020-12-14]. Dostupné z: https://techterms.com/definition/bandwidth

[10] Data Sheet Spirent C1. In: Spirent.com [online]. 2020 [cit. 2020-12-16]. Dostupné z: https://assets.ctfassets.net/wcxs9ap8i19s/50StJiFwpWtUBagDVIIz1T/e455da7d47d2f4c782a9eece9c7df9c5/DS-Spirent-TestCenter-C1.pdf

[11] ParaScope GigE. In: Atecorp.com [online]. 2020 [cit. 2020-12-16]. Dostupné z: https://www.atecorp.com/atecorp/media/pdfs/data-sheets/fetest_parascope-gige_datasheet.pdf

[12] IPerf. In: Iperf.fr [online]. [cit. 2020-12-16]. Dostupné z: https://iperf.fr/iperf-doc.php

[13] Protokoly internetu - HTTP: Kapitola 3. Příkazy protokolu HTTP 1.0 [online]. [cit. 2021-04-23]. Dostupné z: http://www.cs.vsb.cz/grygarek/PS/kotasek/htp03.htm

[14] Protokoly internetu - HTTP: Kapitola 4. Příkazy protokolu HTTP 1.1 [online]. [cit. 2021-04-23]. Dostupné z: http://www.cs.vsb.cz/grygarek/PS/kotasek/htp04.htm

[15] HTTP Methods GET vs POST: The PUT Method [online]. [cit. 2021-04-23]. Dostupné z: https://www.w3schools.com/tags/ref_httpmethods.asp#:~:text=The%20difference%20between%20POST%20and,the%20same%20resource%20multiple%20times.

[16] Understanding the SMTP Protocol: SAMPLE COMMUNICATION [online]. [cit. 2021-4-26]. Dostupné z: https://www.smtp2go.com/blog/understanding-smtp-protocol/

Seznam příloh

Příloha A: Návod k obsluze Spirent TestCenter C1 na úrovni aplikačních protokolů a nastavení pro měření parametrů protokolu HTTP I

Příloha B: Nastavení pro měření protokolu FTP LVII

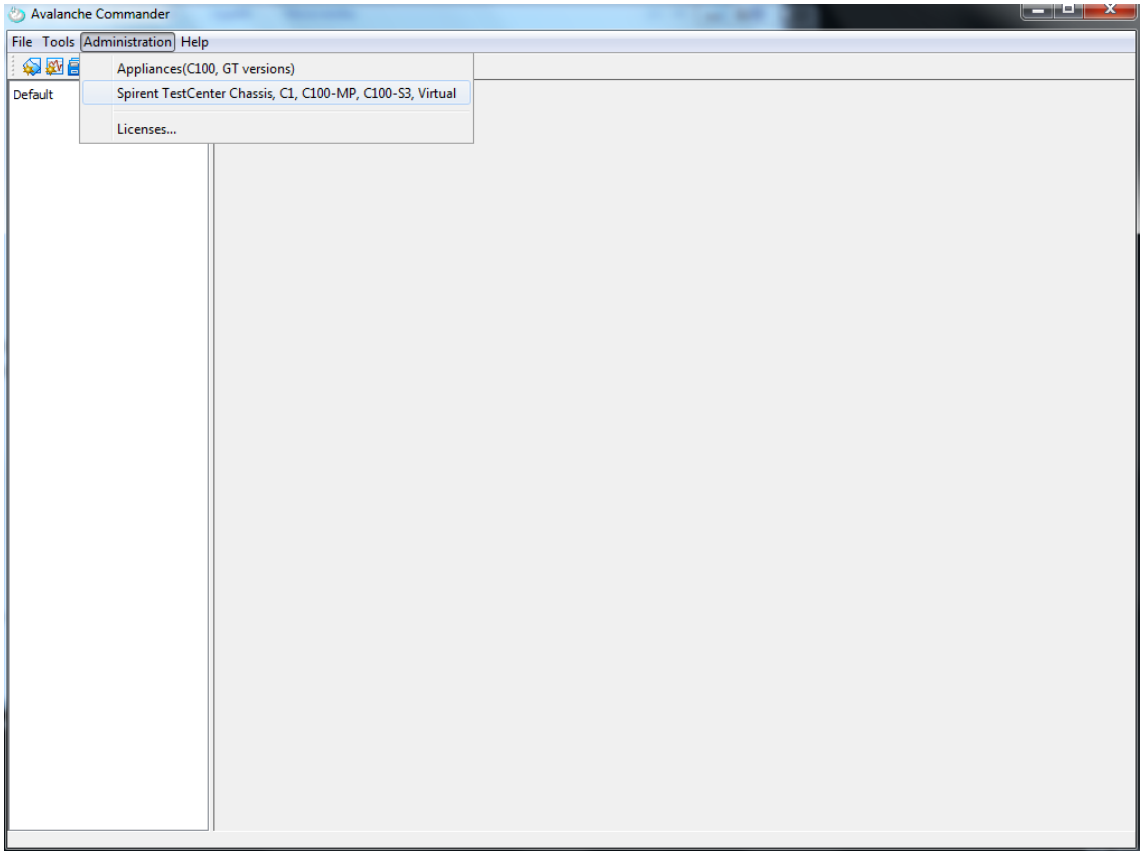
Příloha C: Nastavení pro měření protokolu SMTP LX

Příloha D: Nastavení pro měření několika aplikačních protokolů najednou LXIII

A Návod k obsluze Spirent TestCenter C1 na úrovni aplikačních protokolů a nastavení pro měření parametrů protokolu HTTP

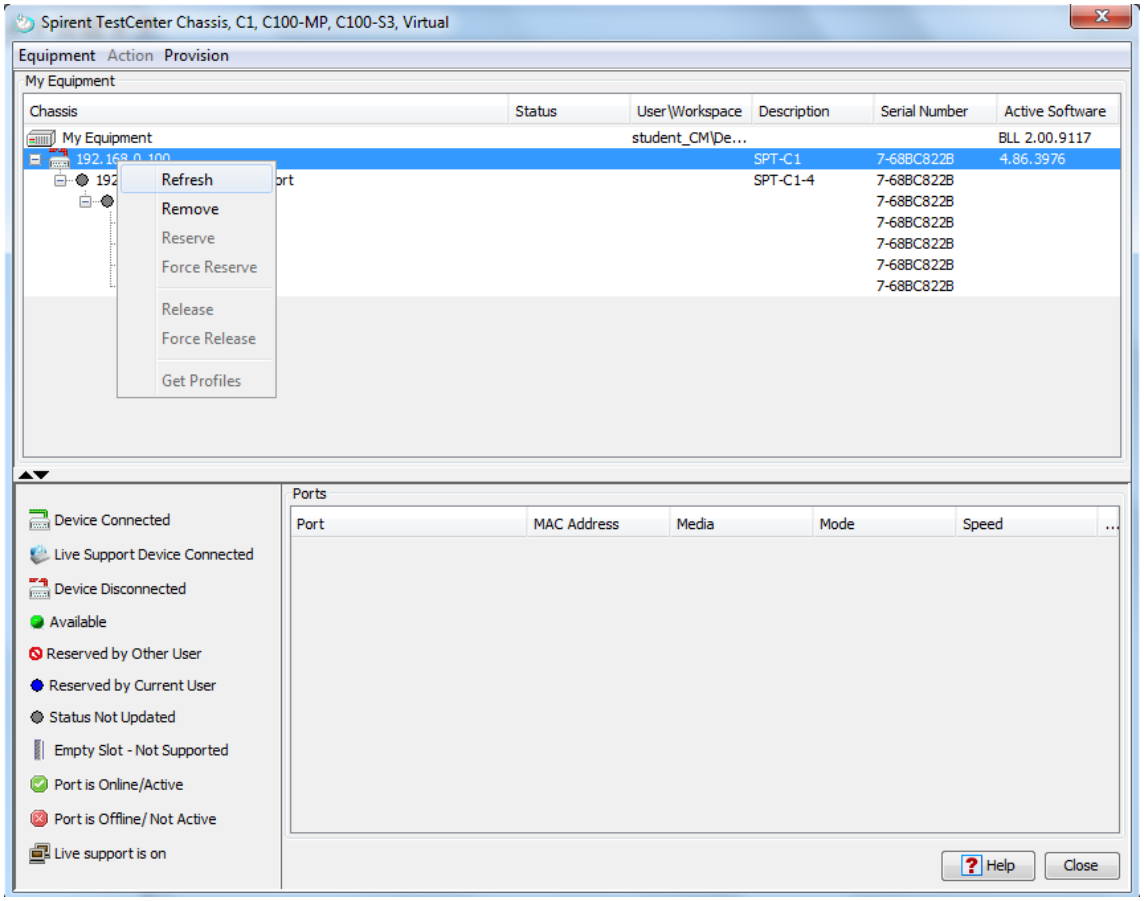
Pro generování datového provozu na úrovní aplikační vrstvy slouží Avalanche Commander, který spustíme kliknutím na program Spirent TestCenter Layer 4-7 Application 4.86.

Jako první však musíme propojit počítač se zařízením Spirent pomocí UTP kabelu. Druhým krokem je propojení zařízení Spirent s testovanou sítí/serverem pomocí dostupných portů, v tomto případě bylo zařízení připojeno do sítě pomocí dvou UTP kabelů, jeden port bude použit ke generování provozu a druhý k simulaci serveru. Po splnění těchto dvou kroků můžeme spustit Avalanche Commander.



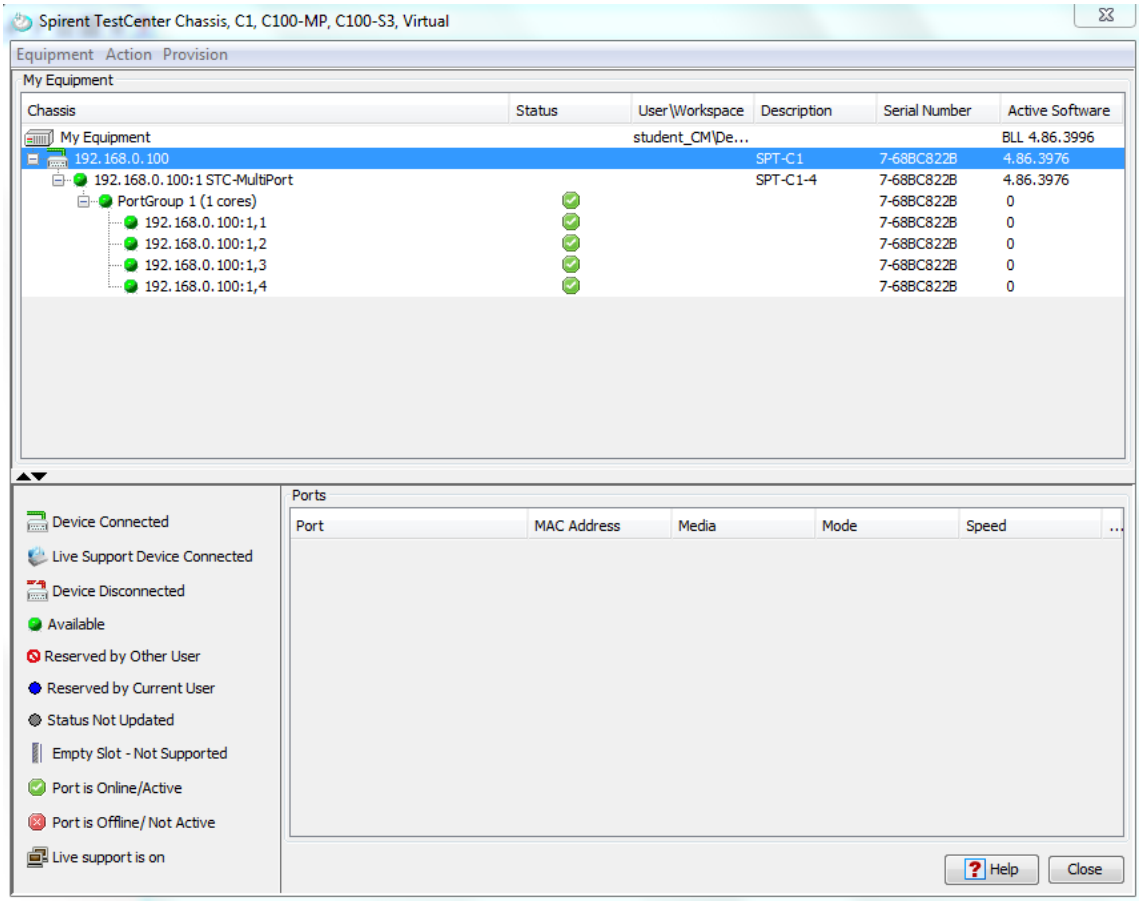
Obrázek A.1: Úvodní obrazovka Avalanche Commanderu

Po spuštění programu musíme jako první kliknout na záložku Administration a v ní otevřít podzáložku Spirent TestCenter Chassis, C1, C100-MP, C100-S3, Virtual. Zde musíme zarezervovat porty potřebné k našemu testování.



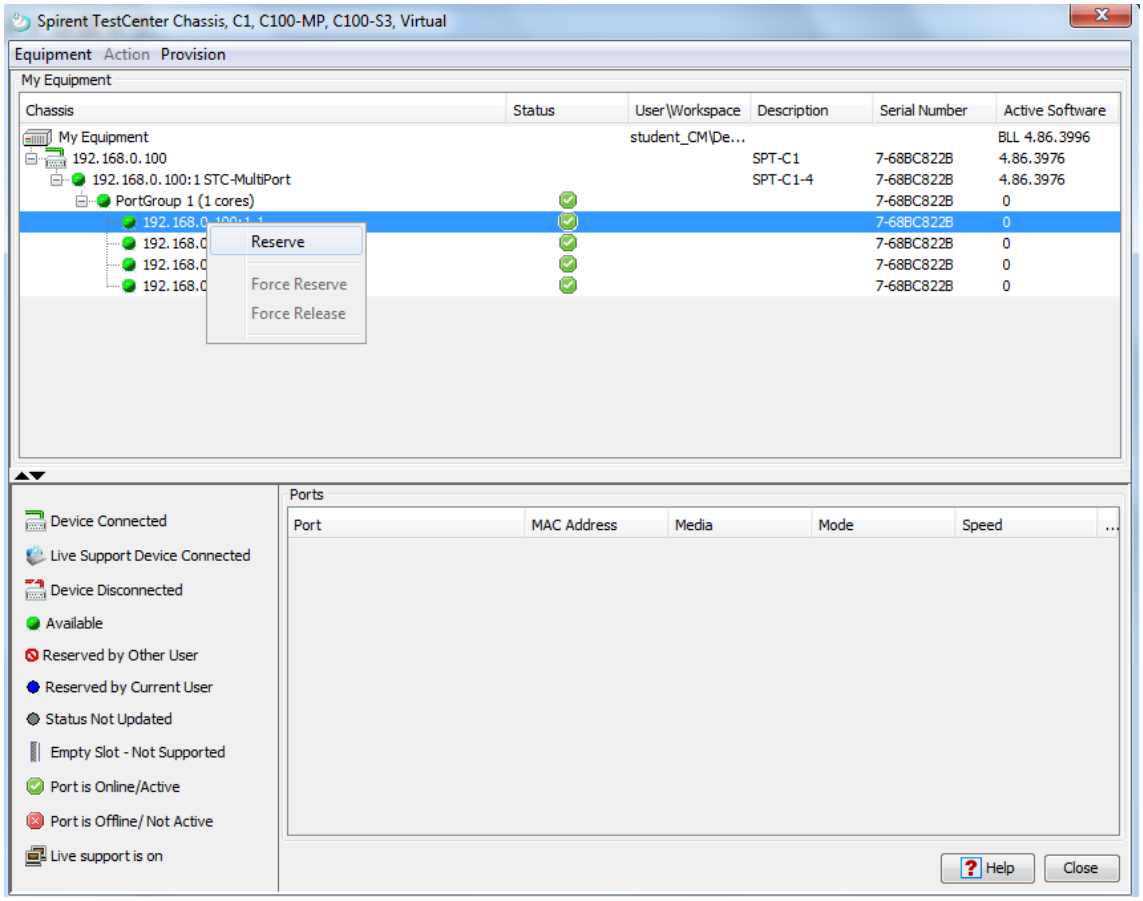
Obrázek A.2: Podzáložka Spirent TestCenter Chassis, C1, C100-MP, C100-S3, Virtual

Pokud je již zařízení Spirent přidané v seznamu tak stačí kliknout pravým tlačítkem myši na IP adresu zařízení a poté na tlačítko Refresh. Poté se zobrazí veškeré aktivní porty daného zařízení (změní barvu ukazatele z šedé na zelenou). V opačném případě je nutné zařízení přidat tlačítkem Equipment v horní liště vlevo a následným stisknutím podzáložky Add Chassis, kde se následně vloží IP adresa přidělená zařízení Spirent.



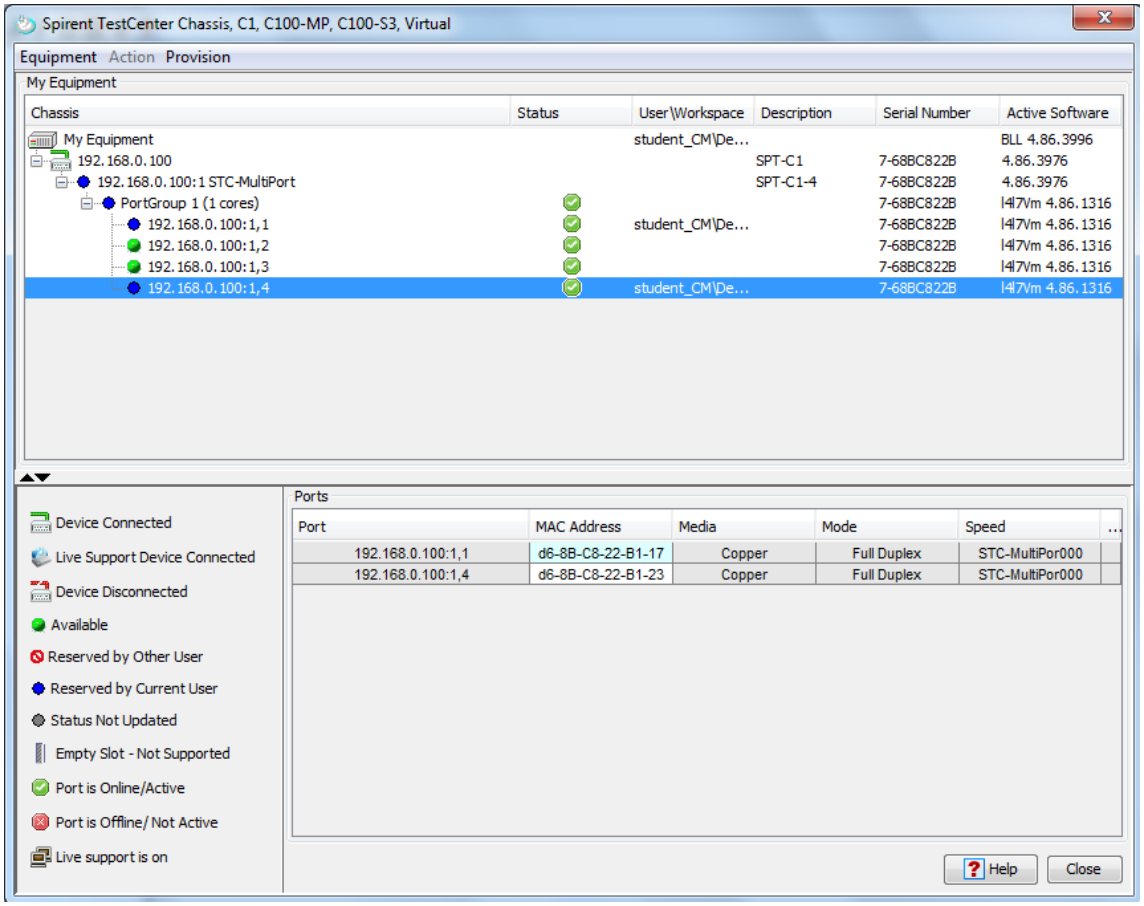
Obrázek A.3: Dostupné porty zařízení Spirent

Na obrázku A.3 vidíme, že zařízení Spirent má dostupné celkem 4 porty. Pokud chceme v testování některý z portů použít, tak jej musíme zarezervovat. To uděláme tak, že pravým tlačítkem klikneme na řádek portu a poté na tlačítko Reserve.



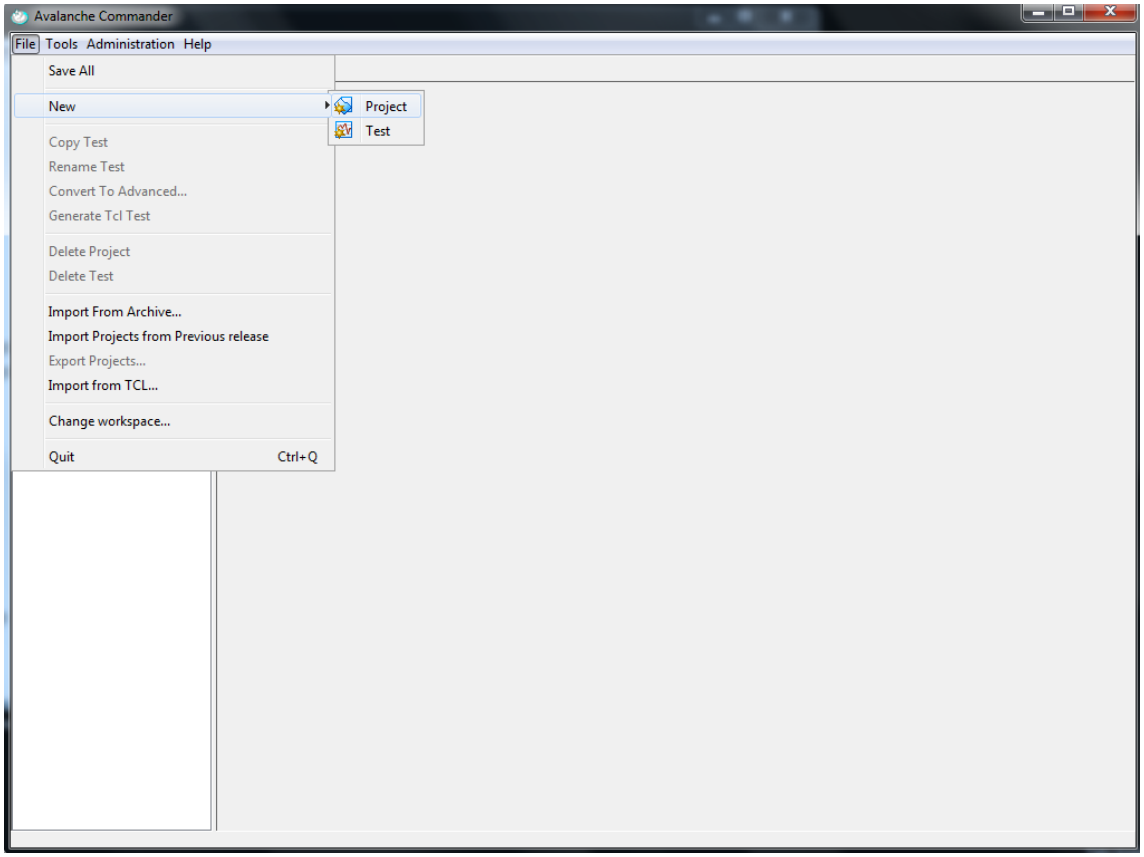
Obrázek A.4: Rezervování portů zařízení Spirent

Jakmile zarezervujeme požadované porty můžeme přejít k samotnému měření, rezervovaný port poznáme změnou zeleného ukazatele na modrý a také tím, že se záznam portu objeví v tabulce v dolní části okna. To můžeme vidět na obrázku A.5.



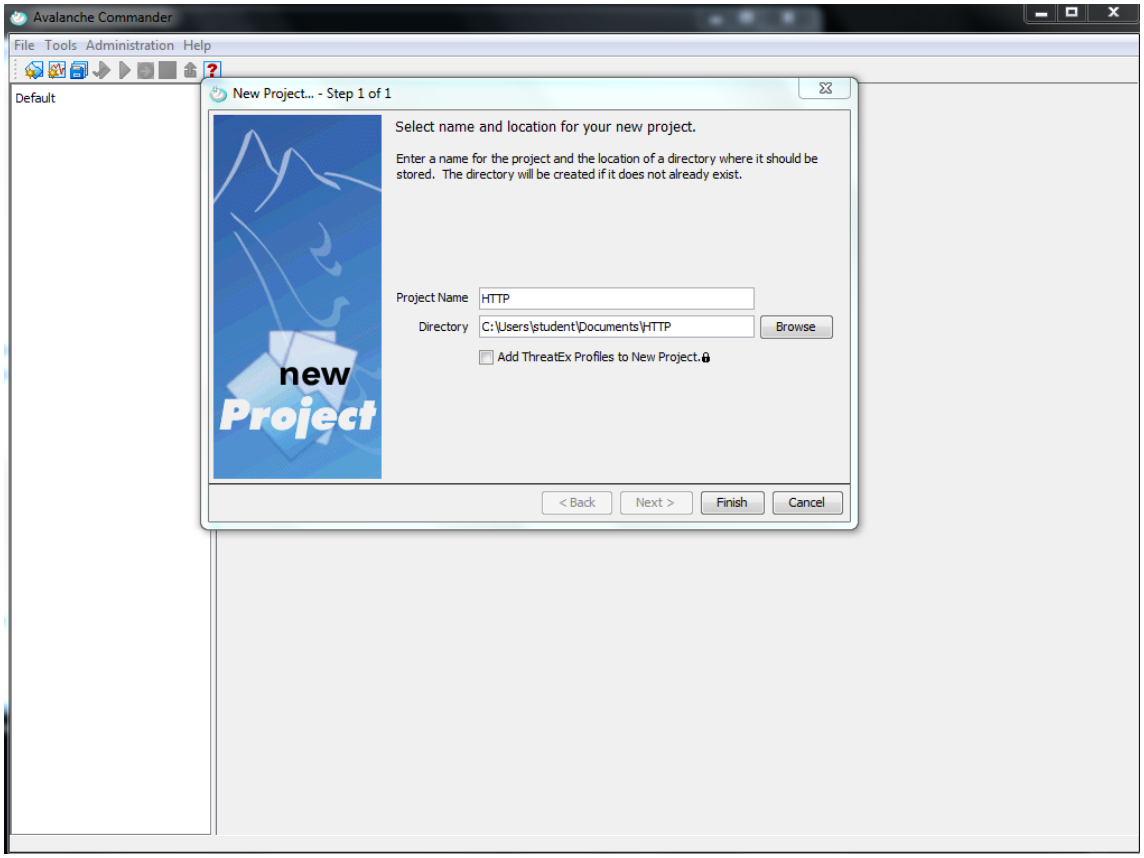
Obrázek A.5: Zarezervované porty 1 a 4

Nyní můžeme zavřít podzáložku Spirent TestCenter Chassis, C1, C100-MP, C100-S3, Virtual a započít měření.



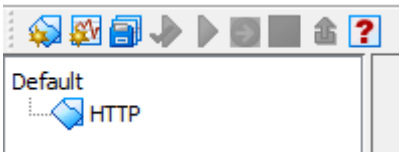
Obrázek A.6: Tvorba projektu - 1

V hlavním okně Avalanche Commanderu klikneme v horní liště na tlačítko File -> New -> Project (obrázek A.6), čímž vytvoříme projekt, který seskupuje následně vytvořené testy.

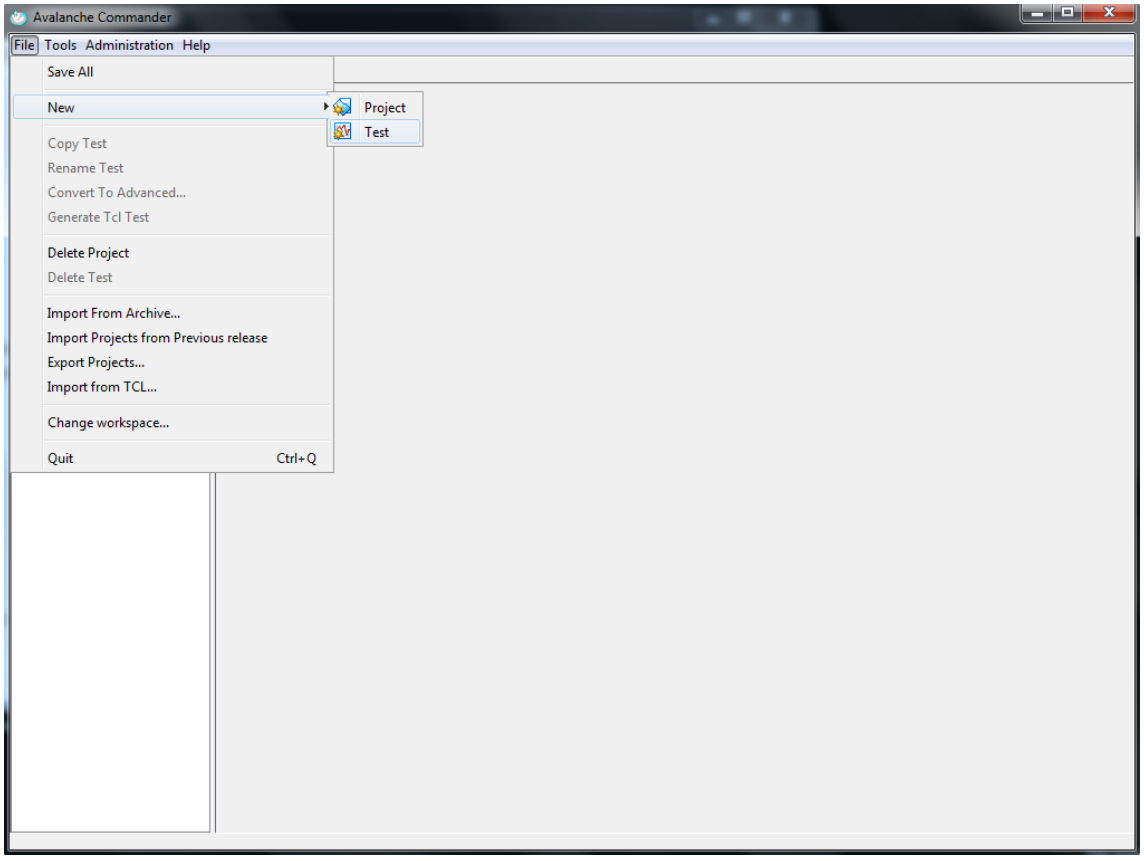


Obrázek A.7: Tvorba projektu - 2

Po provedení předchozího kroku se otevře nové okno, kde musíme nastavit jméno projektu a taktéž můžeme určit, kde se uloží. Po potvrzení našeho nastavení se vytvořený projekt objeví v panelu na levé straně okna.

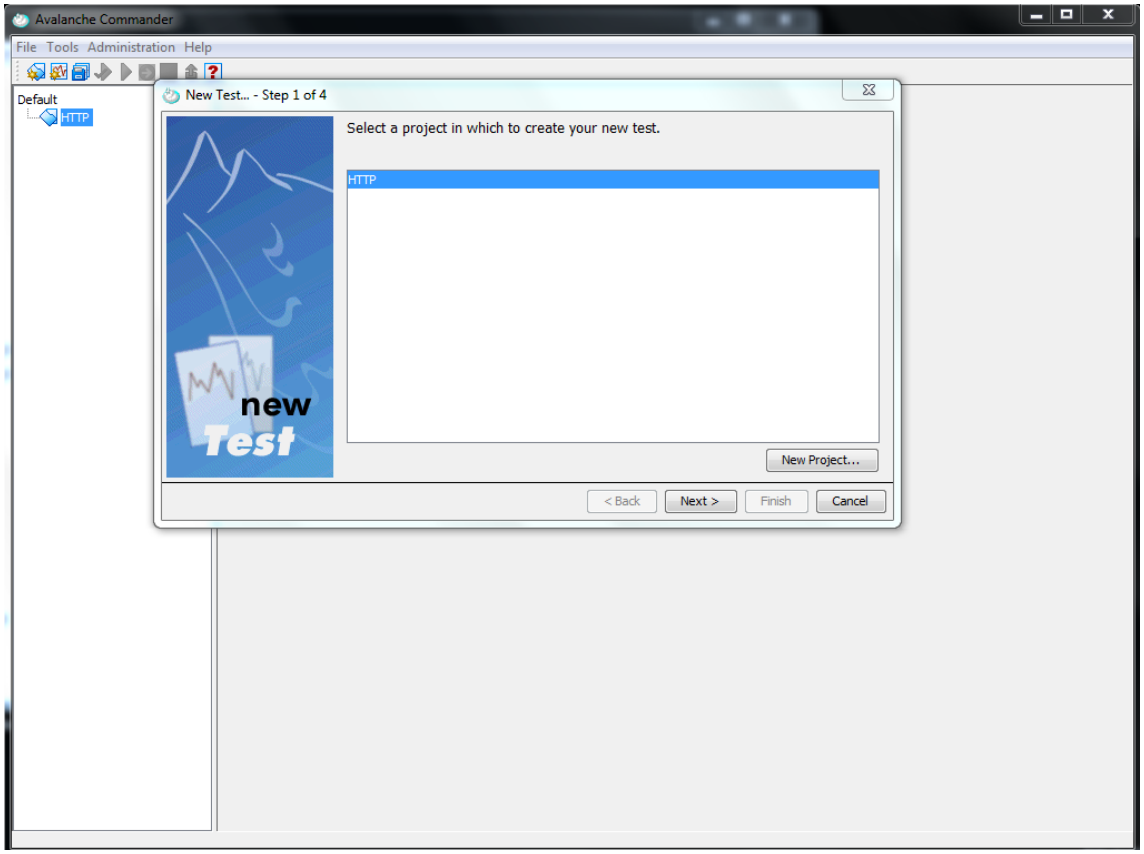


Obrázek A.8: Vytvořený projekt



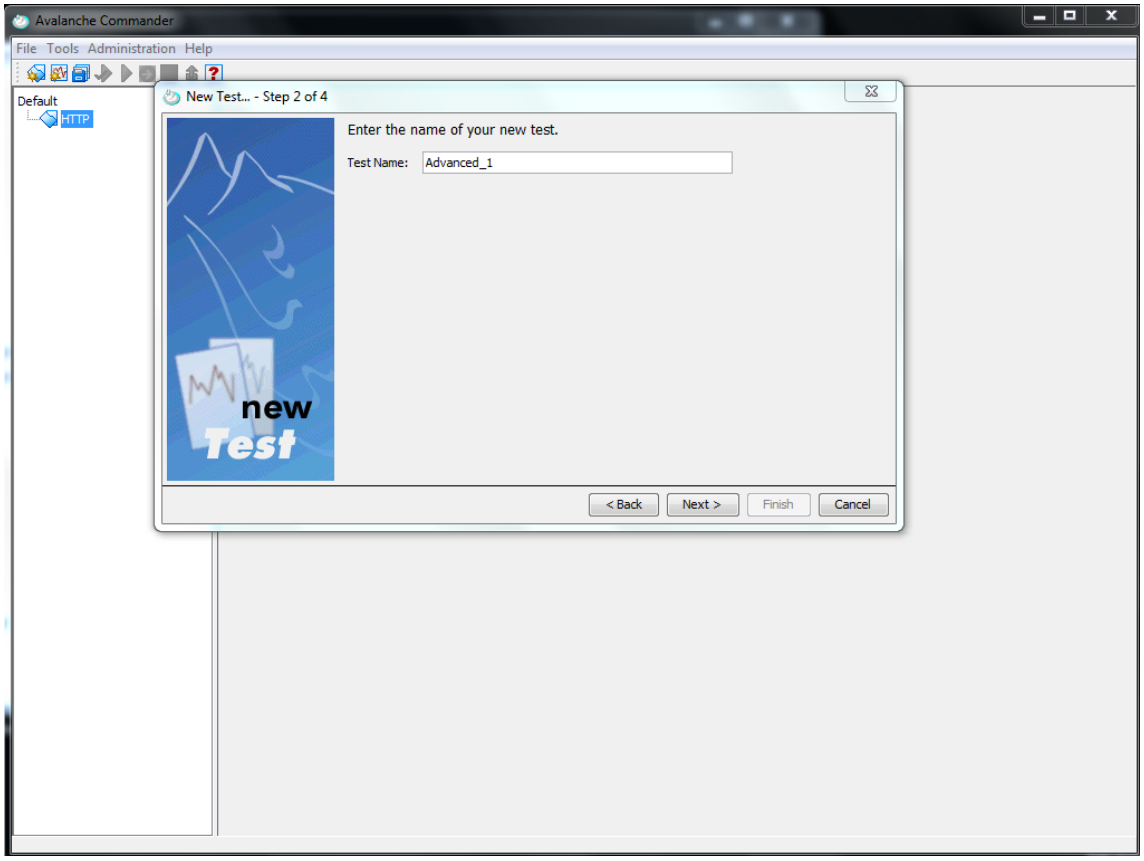
A.9: Tvorba testu - 1

Jakmile je projekt vytvořen, tak můžeme vytvářet jednotlivé testy, postup je podobný jako u vytváření projektu. V horní liště klikneme na tlačítko File -> New -> Test.



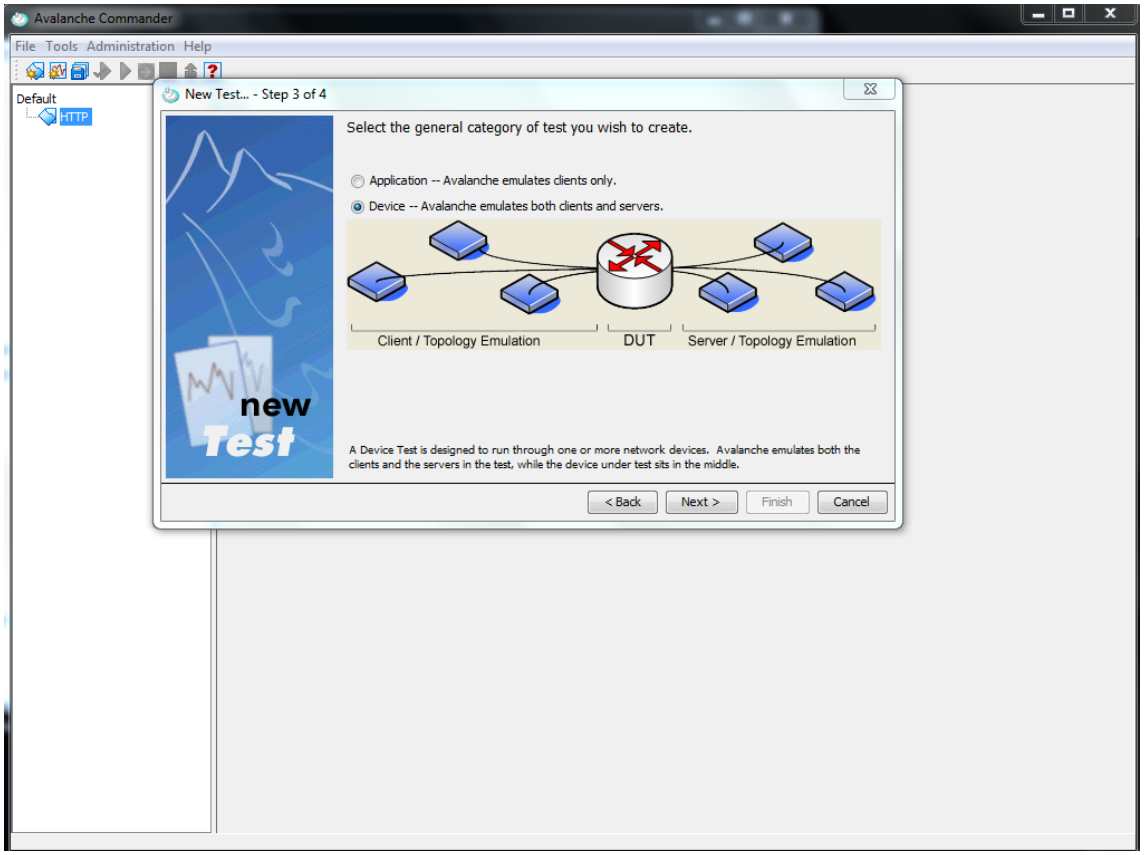
A.10: Tvorba testu - 2

Poté se nám otevře nové okno, kde je nutné test přiřadit k předem vytvořenému projektu. Vybereme předem vytvořený projekt a klikneme na tlačítko Next. Pokud jsme přeskočili předešlý krok, tak je možné vytvořit projekt tlačítkem New Project…



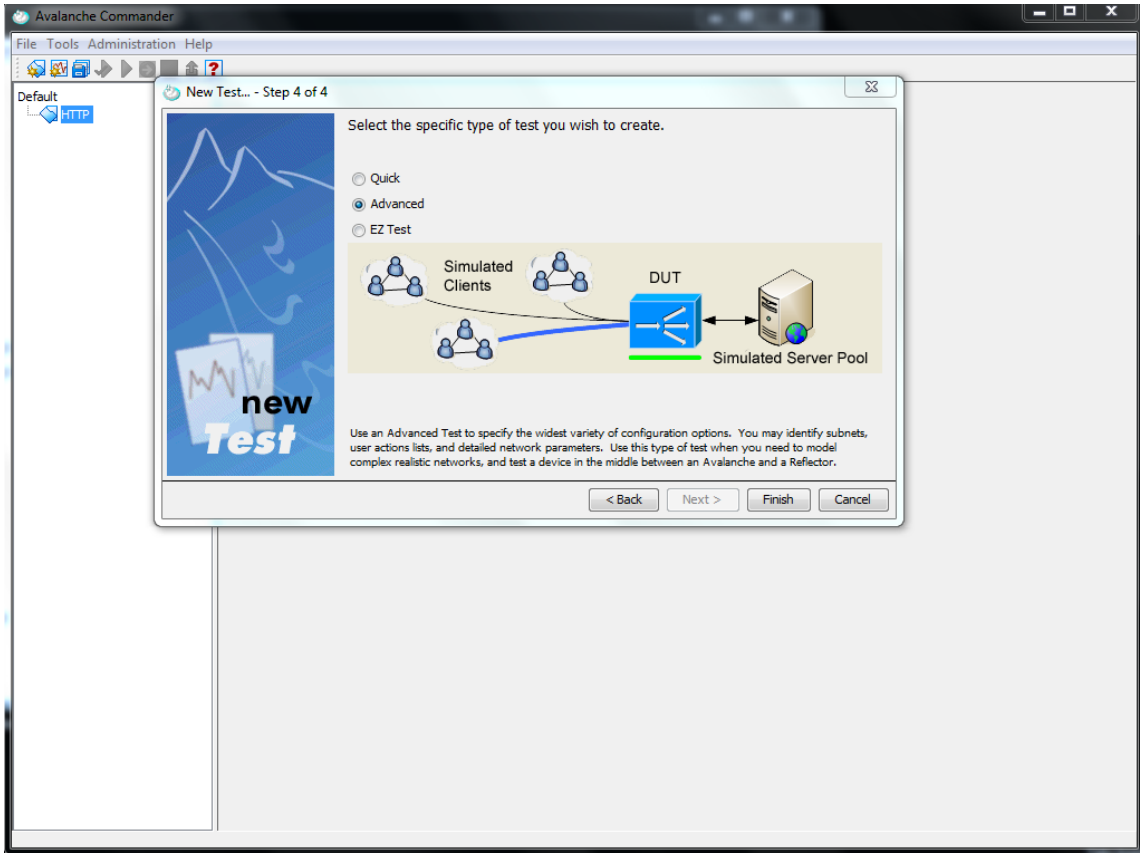
A.11: Tvorba testu - 3

Na další straně se nám otevře okno, kde volíme název testu, jelikož to byl můj první advanced test, tak jsem zvolil název Advanced_1. Pro potvrzení klikneme na tlačítko Next a přejdeme do další části.



Obrázek A.12: Tvorba testu - 4

V dalším okně máme na výběr ze 2 druhů testů. První je Application, kde Spirent slouží pouze jako zátěžový generátor. Druhým je Device, zde zařízení Spirent nejenom generuje datový provoz, ale také umožňuje simulovat server pro některé služby, jako je například web (HTTP), mail (SMTP, POP3) a jiné. Pro potvrzení klikneme na tlačítko Next a přejdeme do další části.

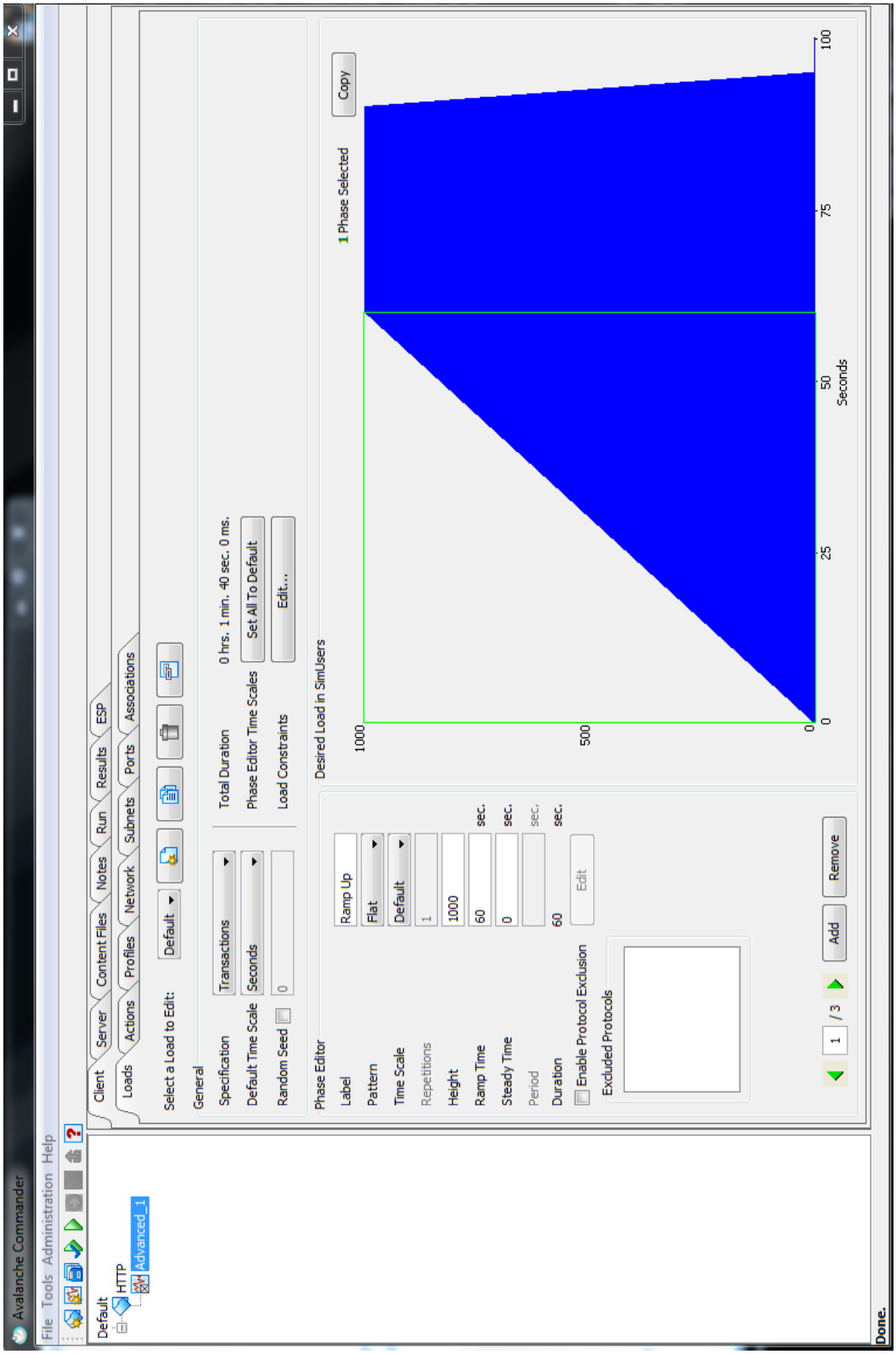


Obrázek A.13: Tvorba testu - 5

Poslední strana nastavení nás žádá o zvolení typu testu, Avalanche Commander umožňuje celkem 3 typy - Quick, Advanced a EZ test. Jednotlivé typy jsou popsány v bakalářské práci, včetně rozdílů a jejich výhod/nevýhod. Já zvolil Advanced, který je z těchto tří nejkomplexnější. Veškeré nastavení je nutné potvrdit tlačítkem Finish.

Po dokončení se objeví námi vytvořený test v levém panelu, kde je spojen s předem vytvořeným projektem. Mimo jiné se otevře nastavení testu, výchozím oknem je záložka Client, podzáložka Loads.

V následující části je postup měření z podkapitoly 4.2.2, jedná se o měření protokolu HTTP, kde je zařízeni Spirent jako klient i server.

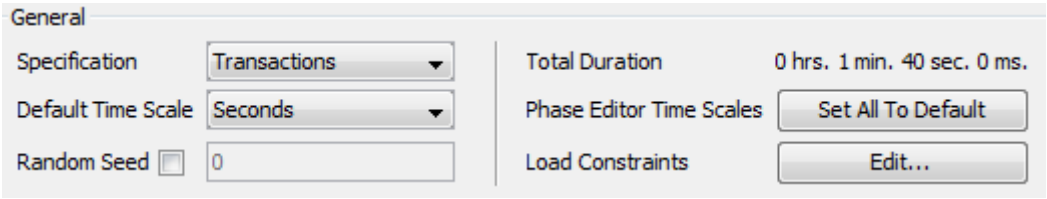


Obrázek A.14: Client - Loads

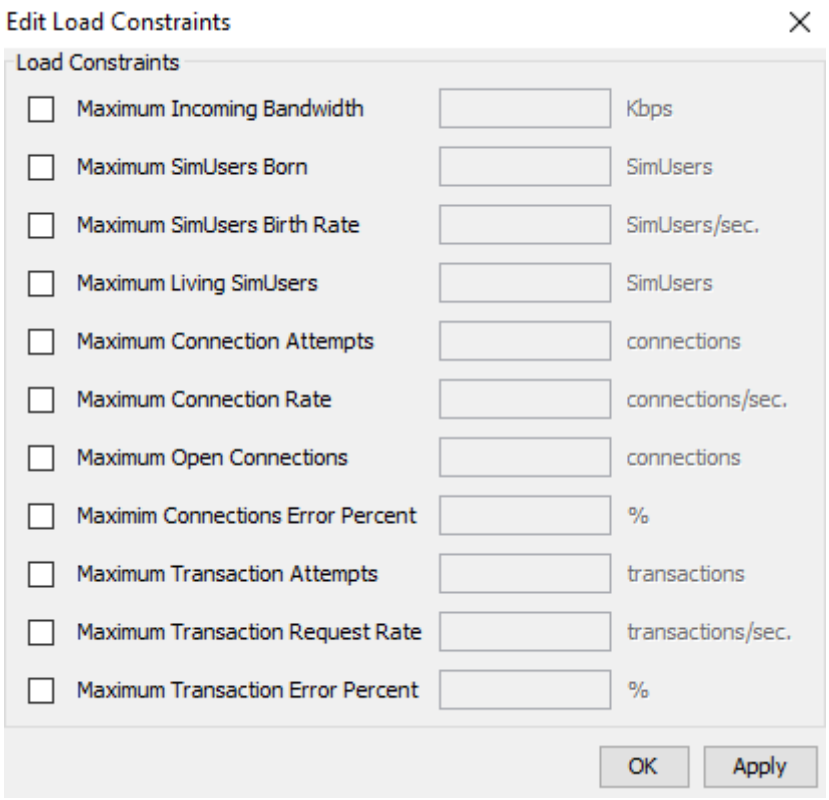
V horní části podzáložky Loads je vyhrazena sekce pro profilování, které umožňuje vytvořit několik různých možností dané záložky. První tlačítko zleva slouží k vybrání profilu. Druhým tlačítkem vytvoříme nový profil. Třetím tlačítkem vytvoříme kopii vybraného profilu. Čtvrté tlačítko slouží k smazání vybraného profilu a pátým tlačítkem můžeme profil přejmenovat. Ve výchozím nastavení máme pouze jeden profil označený názvem Default. Po zvolení profilu můžeme specifikovat nastavení dané záložky.

Obrázek A.15: Sekce pro profilování

O trochu níže je část, která se věnuje základním specifikacím zátěže. V levé části je možno zvolit, jak bude zátěž definovat (Bandwidth, BodyBytes, Connections, Connections/second, Connections/hour, SimUsers, SimUsers/second, SimUsers/hour, Transactions, Transactions/second, Transactions/hour), všechny typy jsou popsány v bakalářské práci s výjimkou Bandwidth a BodyBytes. Bandwidth (propustnost) určuje kolik dat můžeme přenést za daný okamžik. BodyBytes se používá pouze v módu Device, kde server Spirent simuluje protokol HTTP/HTTPS. Pomocí BodyBytes definujeme velikost HTTP odpovědí na generované žádosti. Na dalším řádku volíme časové měřítko testu. Test může probíhat v řádu milisekund, sekund, minut či hodin. V posledním řádku je zaškrtávací políčko Random Seed, pokud ho zaškrtneme, tak poté můžeme zadat seed. Na základě tohoto seedu se generují náhodná čísla, to využíváme u patternu Random. V pravé části pak můžeme vidět celkovou dobu testu. Pod ní je tlačítko, kterým vyresetujeme časová měřítka u všech fází testu do výchozího nastavení. V posledním řádku je tlačítko Edit, tím můžeme omezit zatížení.

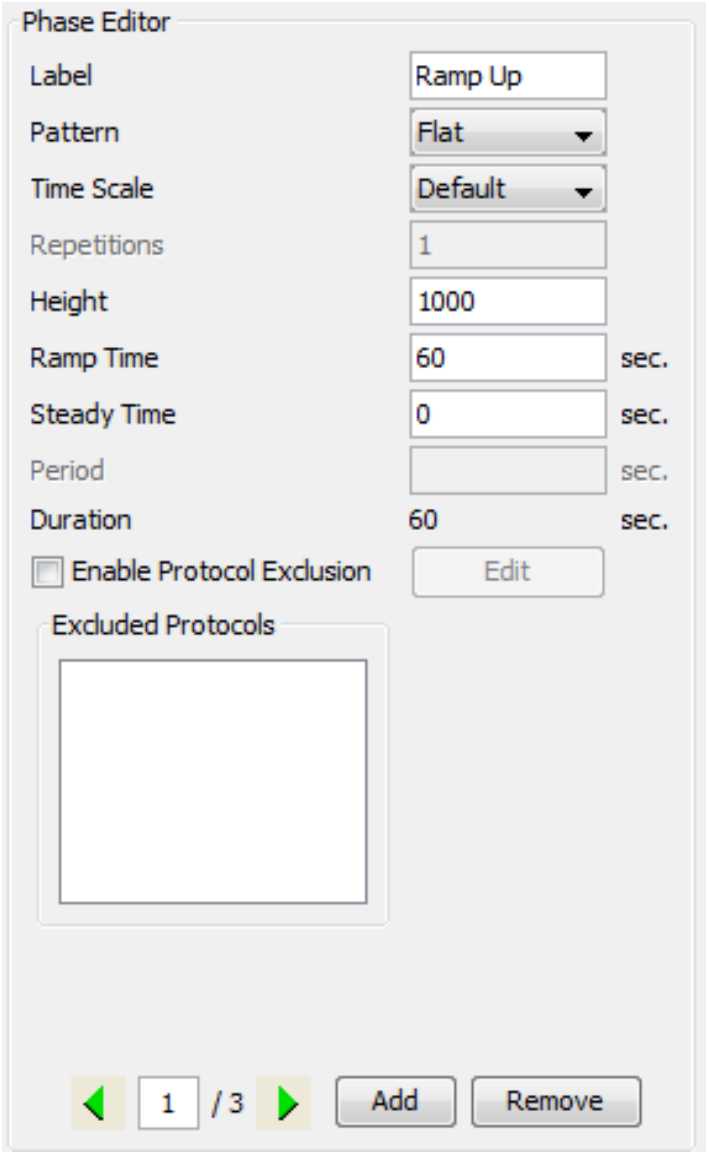


Obrázek A.16: Základní specifikace zátěže

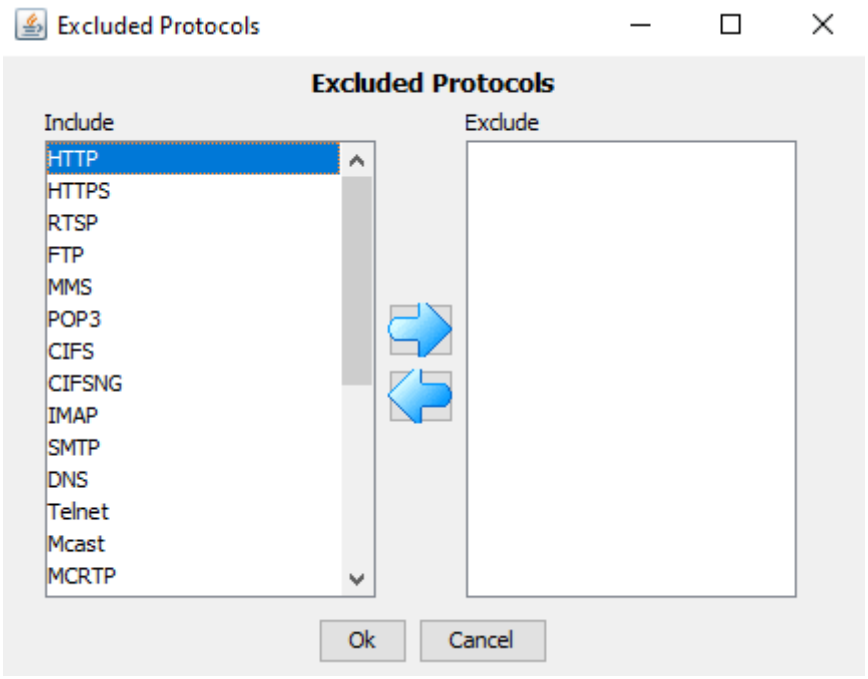


Obrázek A.17: Omezení zatížení

Po základním nastavení je možné tvarovat samotný generovaný provoz. To provádíme v levé dolní části okna. Úplně vlevo dole můžeme přidávat a odebírat fáze testu a následně mezi nimi přepínat pomocí šipek. Jakmile máme vybranou požadovanou fázi, tak se můžeme pustit do samotné úpravy. Jako první můžeme nastavit popis fáze, na dalším řádku vybíráme, jaký tvar (pattern) bude daná fáze mít. Na výběr je plochý, schody, dávky, sinusoida, pilovitý nebo náhodný vzor. Dále můžeme pro každou fázi měnit časové měřítko (milisekundy, sekundy, minuty, hodiny), nebo jej nechat ve výchozím nastavení z předešlé části. Na dalším řádku nastavujeme kolikrát se má daný vzor ve fázi opakovat. O řádek níže nastavujeme, jaké výšky zátěže má fáze dosáhnout. Ramp time určuje, jak rychle se bude zátěž měnit, aby dosáhla výsledné výšky zátěže. Steady time říká, jak dlouho má udržovat maximální nastavenou zátěž v dané fázi. Řádek s označením Period se týká pouze vzoru sinusiod, nastavujeme jím délku periody. Duration pouze ukazuje celkový čas dané fáze, tento parametr nelze měnit. Pokud generujeme více protokolů najednou, můžeme nastavovat jejich generování pouze v určitých fázích testu. K tomu slouží zaškrtávací políčko Enable Protocol Exclusion, který zpřístupníme tlačítko Edit vedle. Po stisknutí tlačítka Edit se otevře nové okno, kde volíme, které protokoly chceme či nechceme v dané fázi použít.

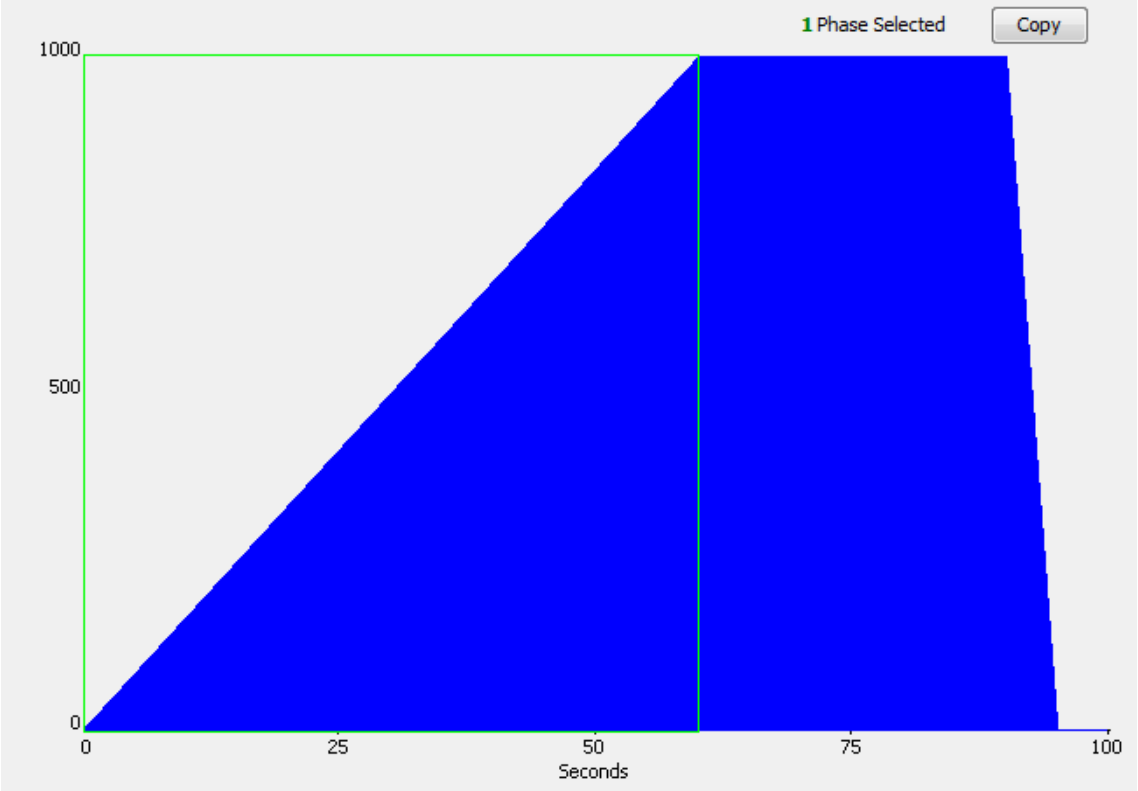


Obrázek A.18: Nastavení fáze

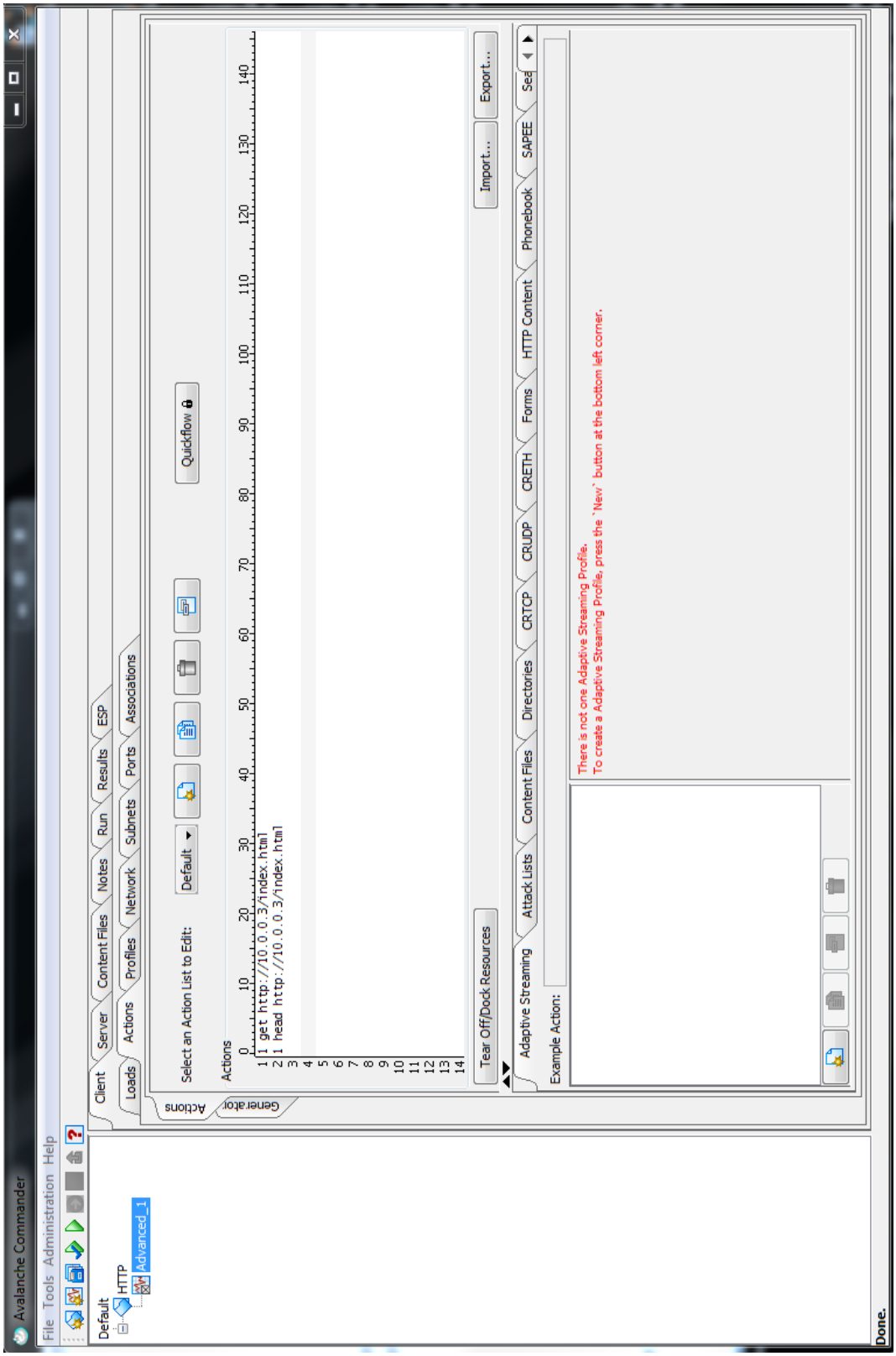


Obrázek A.19: Filtrování aplikačních protokolů

Veškeré výsledky našeho nastavení vidíme real-time v pravé dolní části okna, kde je zobrazen aktuální tvar generované zátěže. Zeleným rámem je označena fáze, kterou momentálně upravuje. Tvar mnou vytvořené zátěže pro toto měření můžeme vidět na obrázku A.20 s tím že zátěž byla specifikována pomocí Transactions.



Obrázek A.20: Tvar generované zátěže (HTTP)



Obrázek A.21: Client - Actions

Podzáložka Actions má další 2 podzáložky a těmi je Actions a Generator. Podzáložka Generator je alternativní verzí podzáložky Actions, já ve svém měření využil podzáložku Actions, a tudíž nebudu popisovat podzáložku Generator. V horní části je opět část vyčleněna profilování, tak jako v podzálože Loads. Zbytek okna je rozdělen na dvě části, v první části vkládáme akce, které chceme generovat. Můžeme taky vložit předem připravené akce tlačítkem Import anebo uložit nyní vytvořené akce tlačítkem Export. Při vytvoření testu jsou zde už některé akce předvytvořené. Druhá část se týká zdrojů pro použité akce, tuto část jsem v žádném měření nevyužil. Můžeme zde například vložit soubory, které chceme následně přenášet protokolem FTP. Pro tento test byly generovány metody HEAD a GET (obrázek A.22).

Kdybychom chtěli generovat jiné HTTP žádosti, tak syntaxe bude totožná s tou na obrázku A.22. S tím že vyměníme typ žádosti. Kromě metody GET a HEAD umí Spirent generovat metody DELETE, PUT a POST. V praxi by jejich syntaxe vypadala následovně.

```
1 HTTPDEL http://10.0.0.3/index.html
```

```
1 PUT http://10.0.0.3/<PUT_FILE="obrazek"
CONTENT_TYPE="image/jpg">
```

```
1 POST http://10.0.0.3/index.html<POST_FILE="obrazek"
CONTENT_TYPE="image/jpg">
```

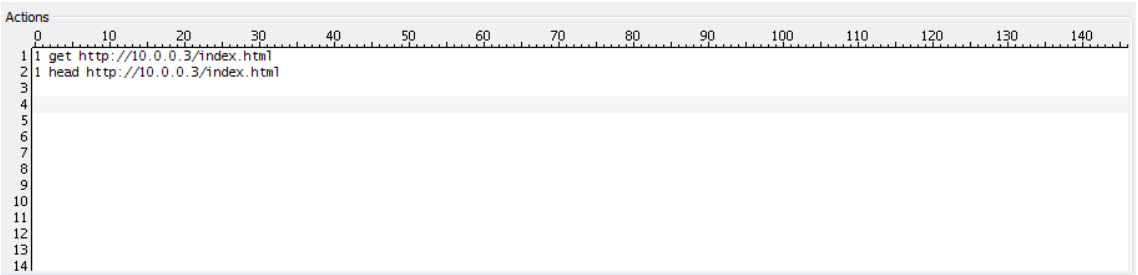
U metod PUT a POST nahráváme obrázek pojmenovaný obrazek, který je ve formátu jpg. Parametrem POST_FILE říkáme který soubor má generátor na server nahrát a CONTENT_TYPE říká, o jaký typ souboru se jedná, popřípadě v jakém je formátu.

Pokud by server vyžadoval autentizace, tak přidáme za příkaz následující parametry, kde místo "jmeno" dáme uživatelské jméno a místo "heslo" naše heslo.

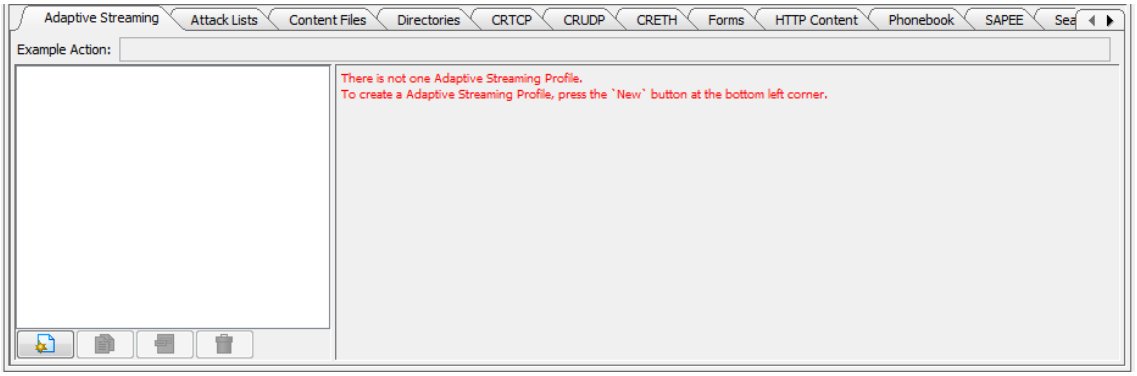
<AUTH: BASIC USER=jmeno PASSWD=heslo>

Pokud proxy server vyžaduje autentizaci, tak přidáme za příkaz následující. Opět místo "jmeno" dáme uživatelské jméno a místo "heslo" vyplníme heslo.

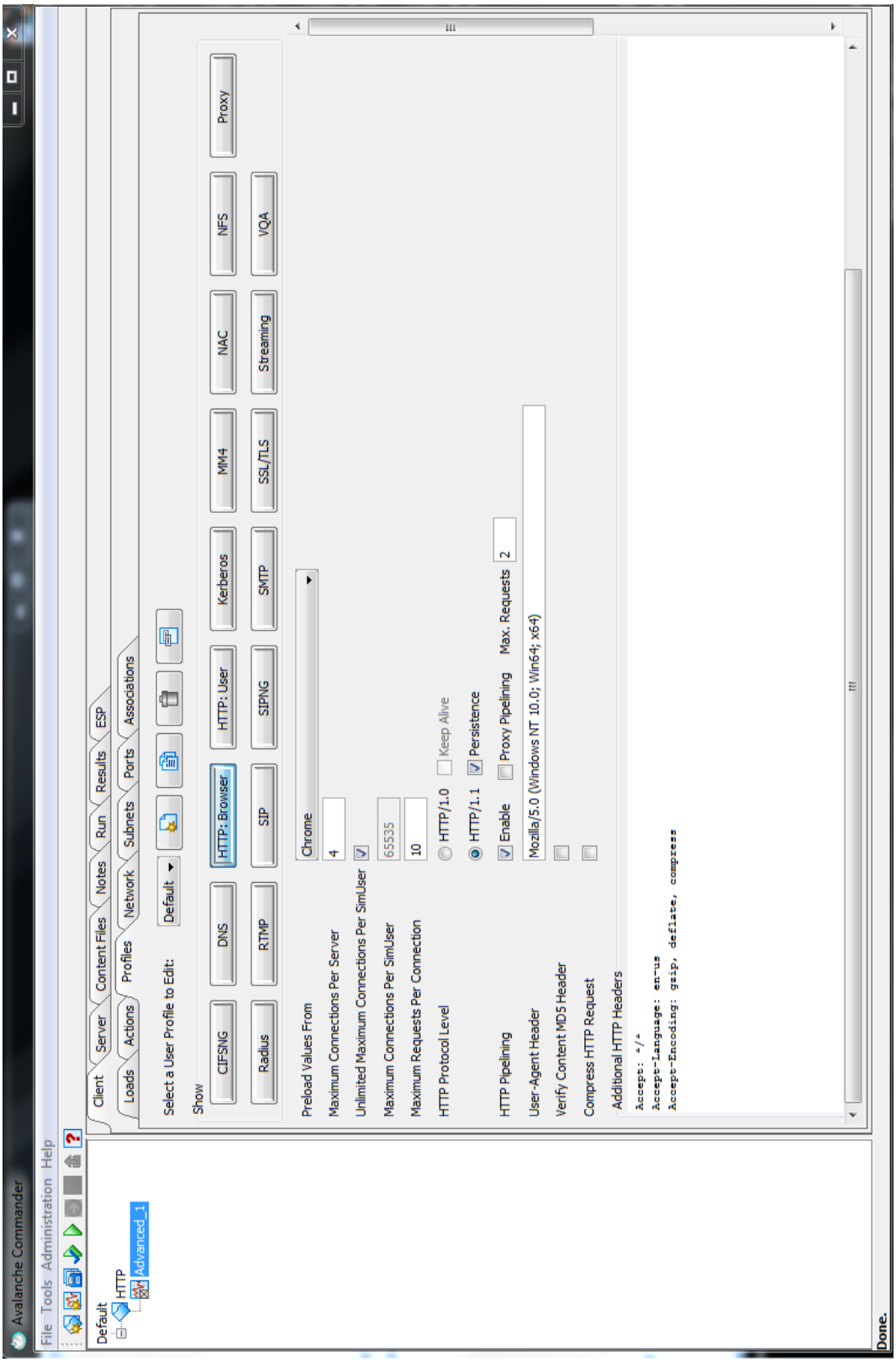
<PROXY AUTH: BASIC USER=jmeno PASSWD=heslo>



Obrázek A.22: Generované akce (HTTP)

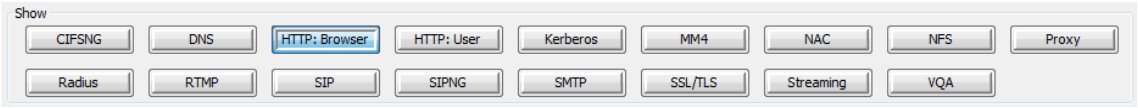


Obrázek A.23: Actions - zdroje



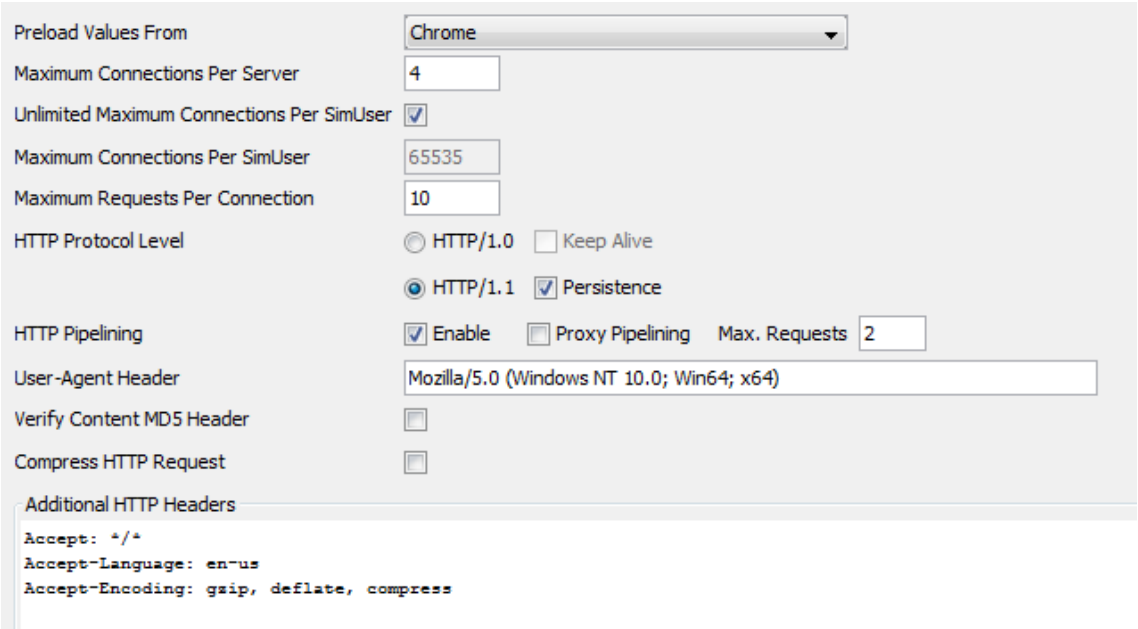
Obrázek A.24: Client - Profiles

V první části podzáložky Profiles je jako obvykle možnost profilování. V další části jsou tlačítka, kterými otvíráme nastavení chování klienta pro daný aplikační protokol. Protokol HTTP, který generuji v tomto testu můžeme upravit rozkliknutím tlačítek HTTP:Browser a HTTP:User. Každé tlačítko má kompletně jiné nastavení, a tudíž budu popíšu pouze tyto dvě.



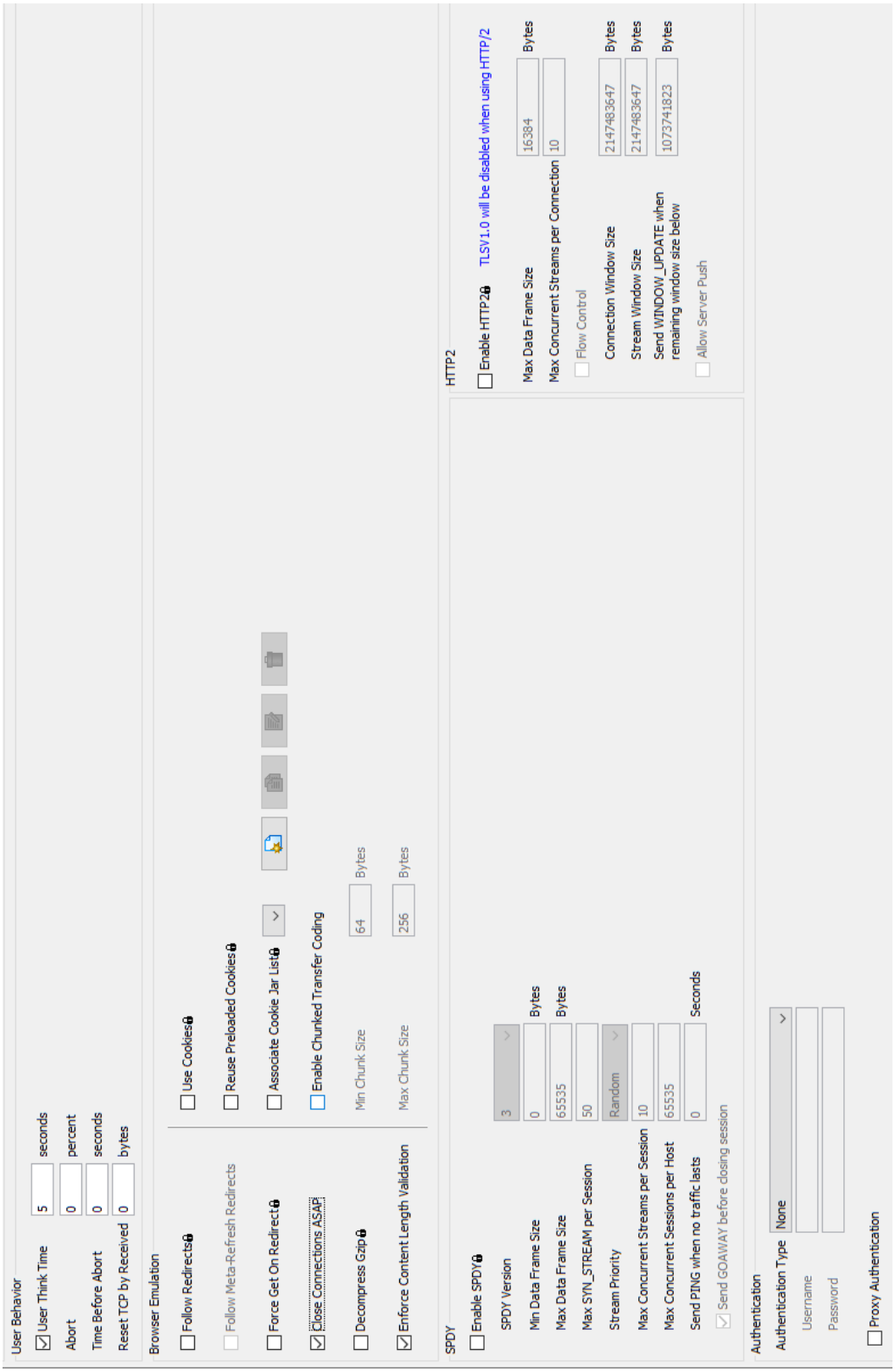
Obrázek A.25: Volba protokolu

Po otevření HTTP:Browser můžeme zvolit jaký typ prohlížeče budou simulovaní klienti používat (Chrome, Edge, Internet Explorer, Firefox, Safari). Dále je možné nastavit například kolik může mít maximálně otevřených spojení se serverem, verzi protokolu HTTP, aktivování pipeliningu a další. Pro tento test bylo nastaven HTTP:Browser podle obrázku A.26.



Obrázek A.26: HTTP:Browser

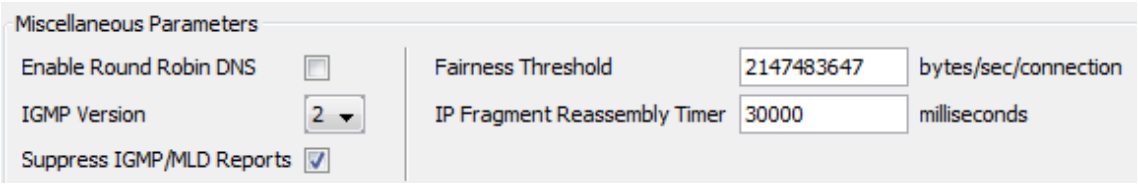
V rámci HTTP:User můžeme nastavit chování simulovaných uživatelů, jak dlouho mají přemýšlet, kolik procent provozu mají zahodit a další. Dále zde lze upravit chování prohlížeče, například, že má ukončovat spojení, jakmile to bude možné, nebo používat Cookies. Dále je zde možnost aktivovat protokol SPDY či HTTP2 a nastavit jejich parametry. Mimo jiné lze nastavit s jakými údaji se bude strana klienta autentizovat u testovaného serveru (v tomto případě by to byl HTTP) nebo Proxy serveru. Pro tento test bylo nastaven HTTP:User podle obrázku A.27.



Obrázek A.27: HTTP:User

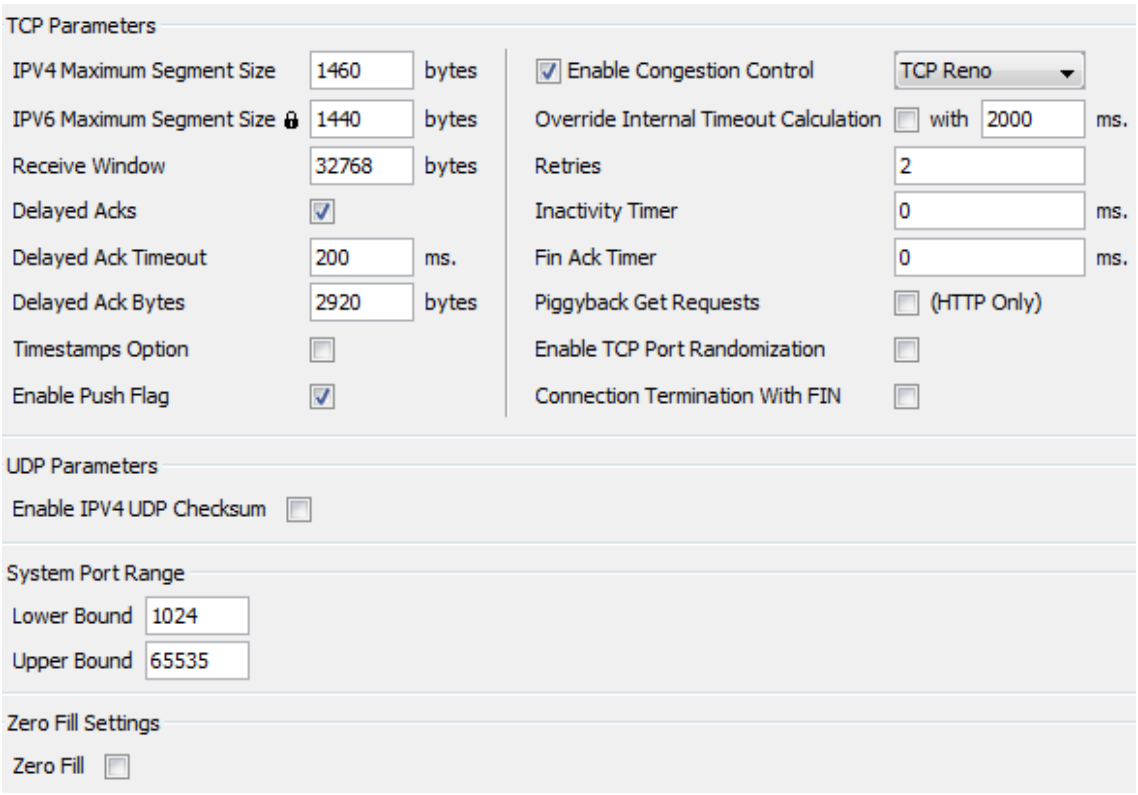
Obrázek A.28: Client - Network

V první části podzáložky Network je možné nastavit různé síťové parametry, jako je například verze IGMP.



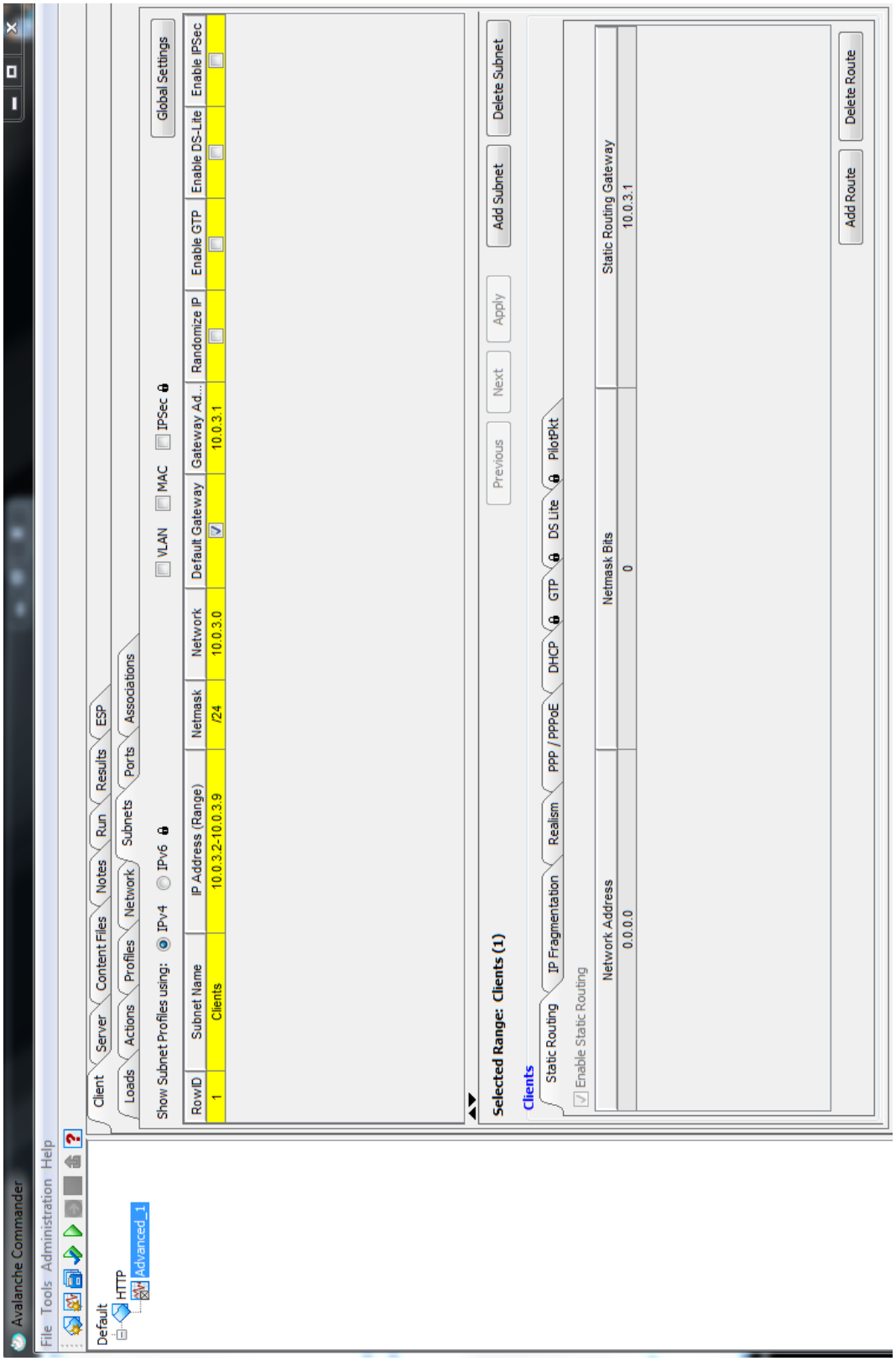
Obrázek A.29: Různé síťové parametry

V další části nastavujeme parametry pro transportní protokoly. U protokolu TCP je to například maximální velikost segmentu pro IPv4/6 nebo ukončování spojení na základě příznaku FIN. U protokolu UDP máme jedinou možnost a tou je aktivace kontrolního součtu. Dále je možné změnit rozsah portů, které strana klienta využívá. Ve výchozím nastavení je to celý rozsah (kromě dobře známých portů) tedy 1024-65535. Poslední zde je možnost aktivovat vyplňování nedůležitých dat v paketech nulami, tím lze zrychlit kontrolní součet.



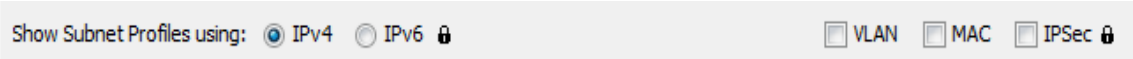
Obrázek A.30: Nastavení parametrů transportní vrstvy

Pro tento test bylo ponecháno výchozí nastavení, které můžeme vidět na obrázku A.29 a A.30.



Obrázek A.31: Client - Subnets

Tato podzáložka slouží k přidání podsítě, ze které bude generován provoz. V horní části volíme typ síťového protokolu mezi IPv4 a IPv6, dále můžeme aktivovat přidání doplňujících parametrů o VLAN, MAC či IPSec.



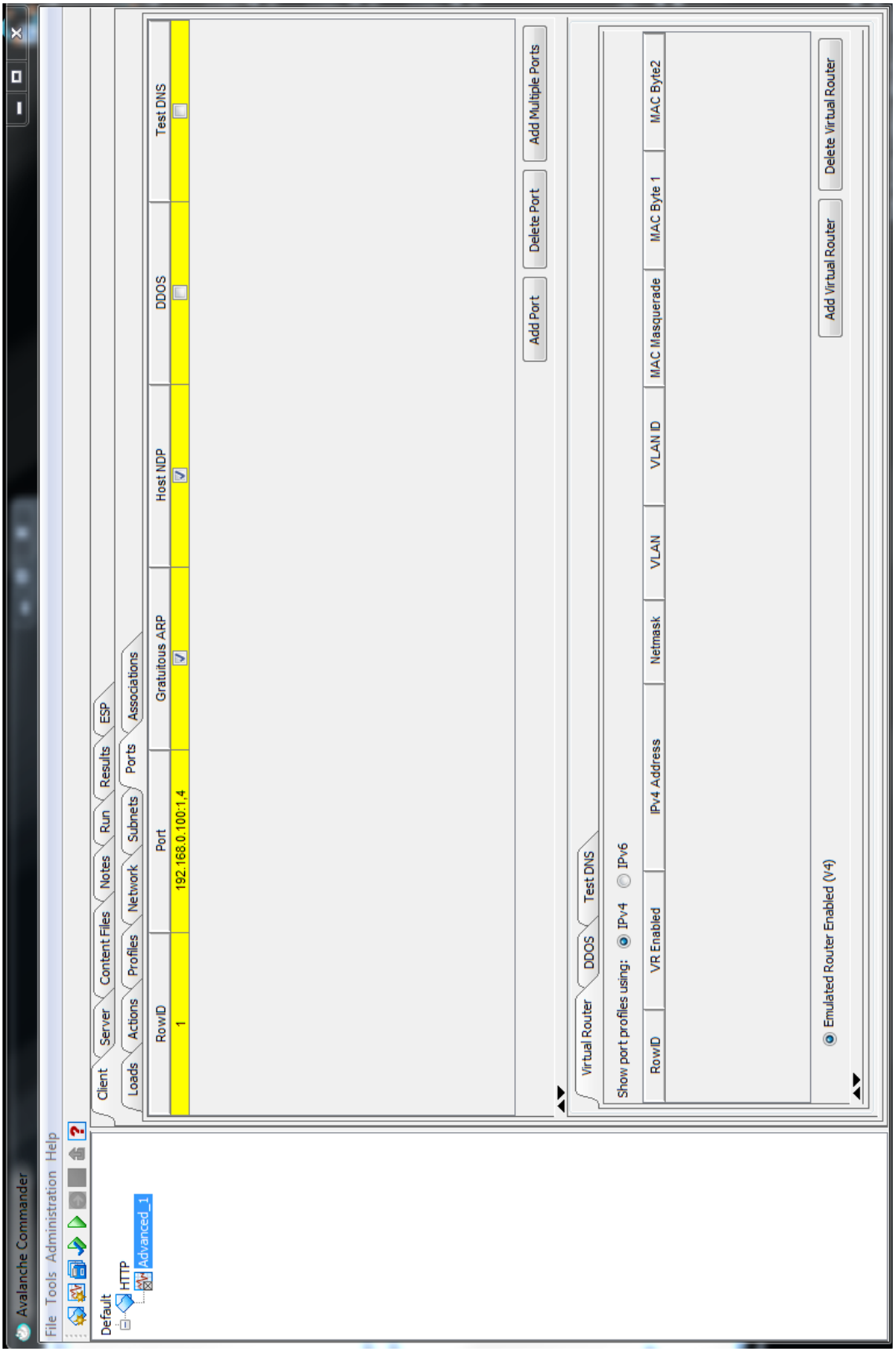
Obrázek A.32: Volba síťového protokolu a doplňujících parametrů

Poté klikneme na tlačítko Add Subnet, čímž přidáme novou podsíť. Následně se nám podsíť objeví v tabulce, většinu parametrů lze změnit tak, že na něj dvakrát klikneme a jednoduše přepíšeme a potvrdíme klávesou enter. Výsledný záznam lze vidět na obrázku A.33. Nejdůležitějším políčkem je IP Address (Range), kde definujeme, z jakých IP adres bude provoz generován. U tohoto testu jsem pojmenoval podsíť jako Clients, nastavil rozsah IP adres na 10.0.3.2-10.0.3.9, výchozí brána byla nastavena na 10.0.3.1.

RowID	Subnet Name	IP Address (Range)	Netmask	Network	Default Gateway	Gateway Ad...	Randomize IP	Enable GTP	Enable DS-Lite	Enable IPSec
1	Clients	10.0.3.2-10.0.3.9	/24	10.0.3.0	☑	10.0.3.1	☐	☐	☐	☐

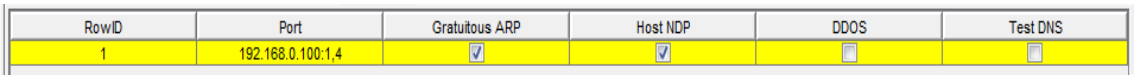
Obrázek A.33: Záznam podsítě - Client

V dolní části okna můžeme upravovat další síťové parametry jako jsou například záznamy o statickém směrování. Této části jsem ve svých měření nevyužil.



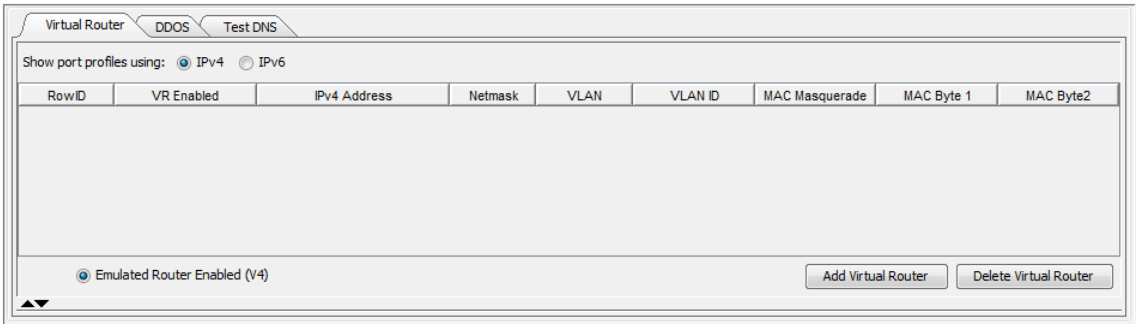
Obrázek A.34: Client - Ports

V této záložce je nutné přidat port zařízení Spirent kterým je daná strana připojena do sítě. To uděláme tlačítkem Add Port. Pro mé měření byly vybrány porty 1 a 4, kde pro stranu klienta využívám port 4. Ve sloupci Port na obrázku A.35 vidíme 192.168.0.100:1,4, kde 192.168.0.100 je IP zařízení Spirent, 1 za IP značí skupinu portů (na našem zařízení je pouze jedna skupina) a číslo 4 značí samotný port. Dále je zde možné aktivovat NDP a ARP protokol či testování DNS a DDoS. Distributed Denail of Service je však dostupné pouze s rozšiřující licencí.

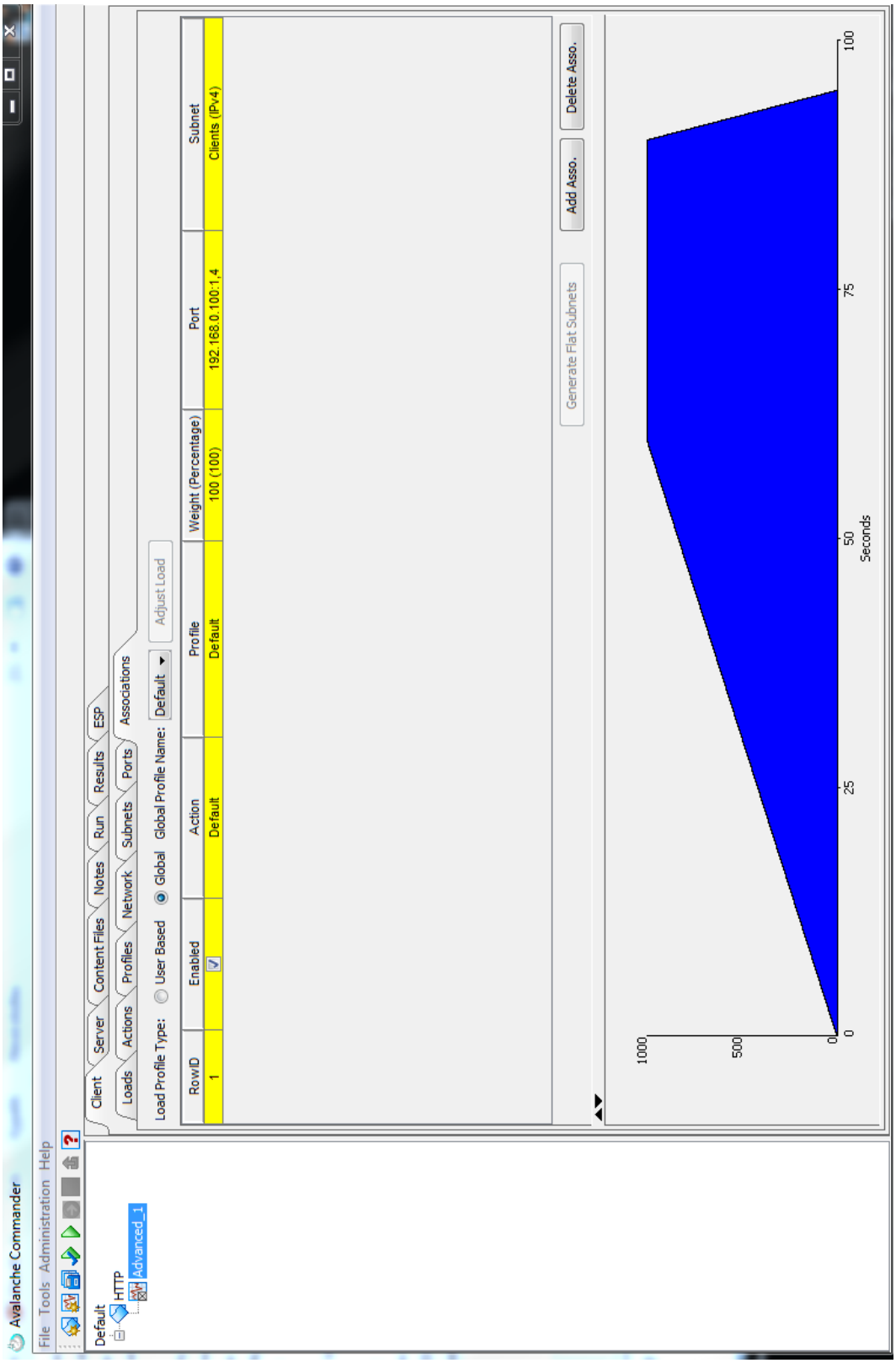


Obrázek A.35: Záznam portu - Client

V dolní části se pak konfigurují doplňující virtuální router, DDoS či DNS. Dolní část záložky Ports jsem ve svém měření nevyužil.

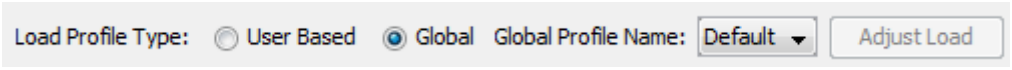


Obrázek A.36: Doplňující nastavení portů



Obrázek A.37: Client - Associations

V této záložce dochází k propojení všech našich předešlých nastavení. Pokud jsme nevyužili možnosti profilování, tak jednoduše všude necháme výchozí nastavení (Default) a pouze stačí kliknout na tlačítko Add Associations vpravo uprostřed obrazovky. Úplně nahoře můžeme nastavit profil zátěže.



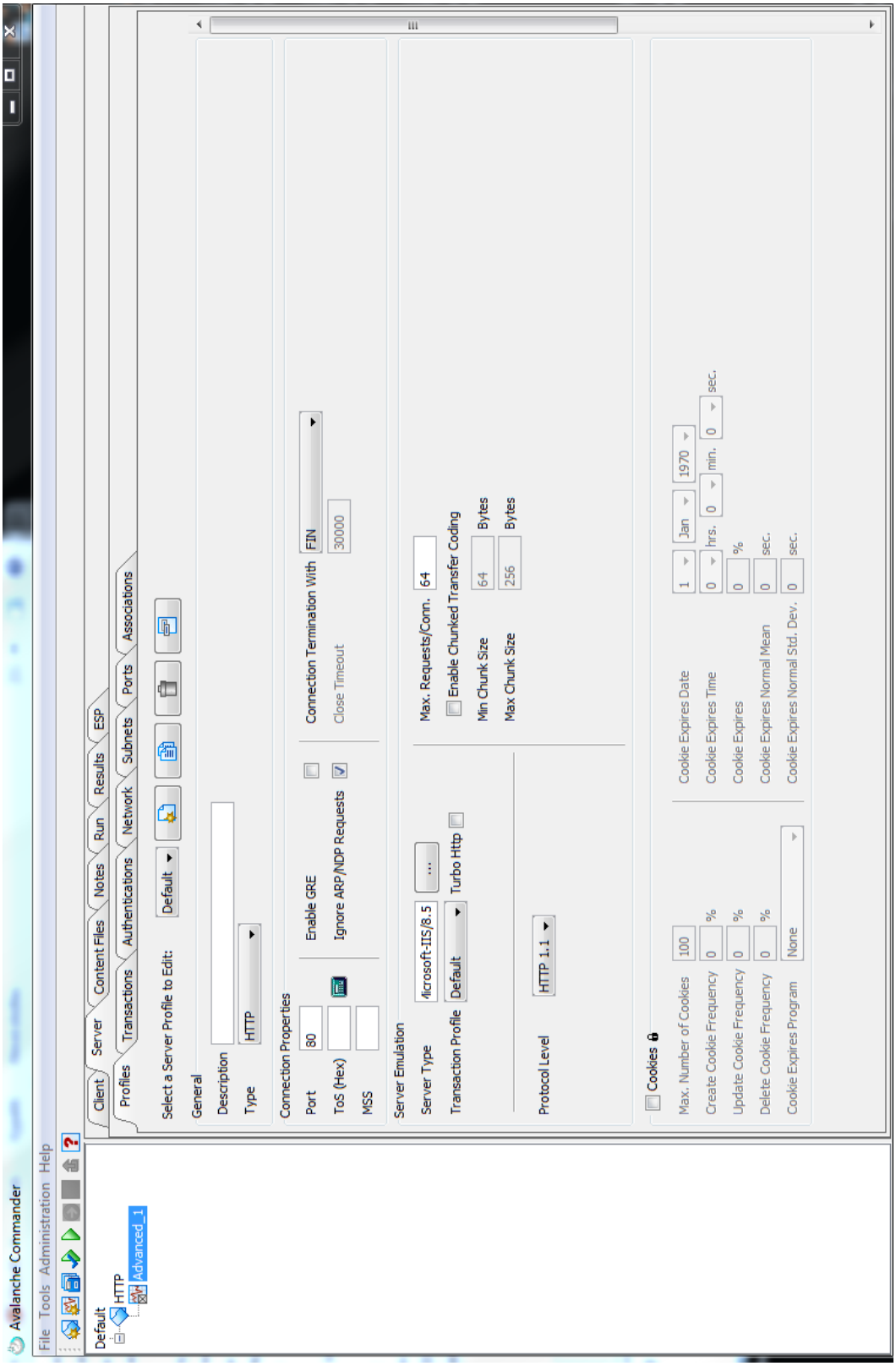
Obrázek A.38: Nastavení profilu zátěže

Pokud bychom chtěli upravit náš záznam, například změnit profil záložky Actions, tak pouze klikneme v daném sloupci na záznam a vyjede nám seznam všech dostupných profilů dané záložky.

RowID	Enabled	Action	Profile	Weight (Percentage)	Port	Subnet
1	☑	Default	Default	100 (100)	192.168.0.100:1,4	Clients (IPv4)

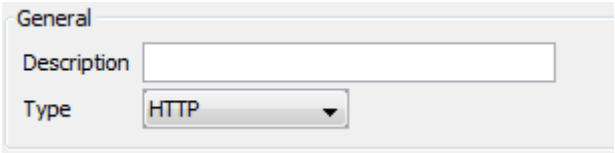
Obrázek A.39: Vytvořený záznam spojených podzáložek záložky Client

Tímto máme na straně klienta hotovo a můžeme se přesunout na stranu serveru.



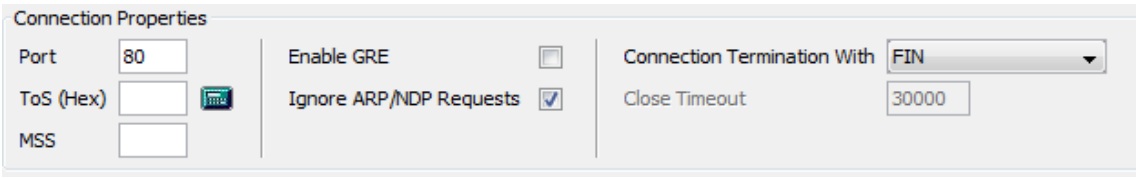
Obrázek A.40: Server - Profiles

První záložka slouží k nastavení typu serveru a jeho parametrů. Úplně nahoře je možnost profilování, neliší se od toho z podzáložek záložky Client. Pod tím je obecné nastavení serveru, v prvním řádku můžeme nastavit popisek serveru. Ve druhém řádku volíme ze seznamu protokol, který bude server simulovat, v tomto případě se jedná o protokol HTTP.



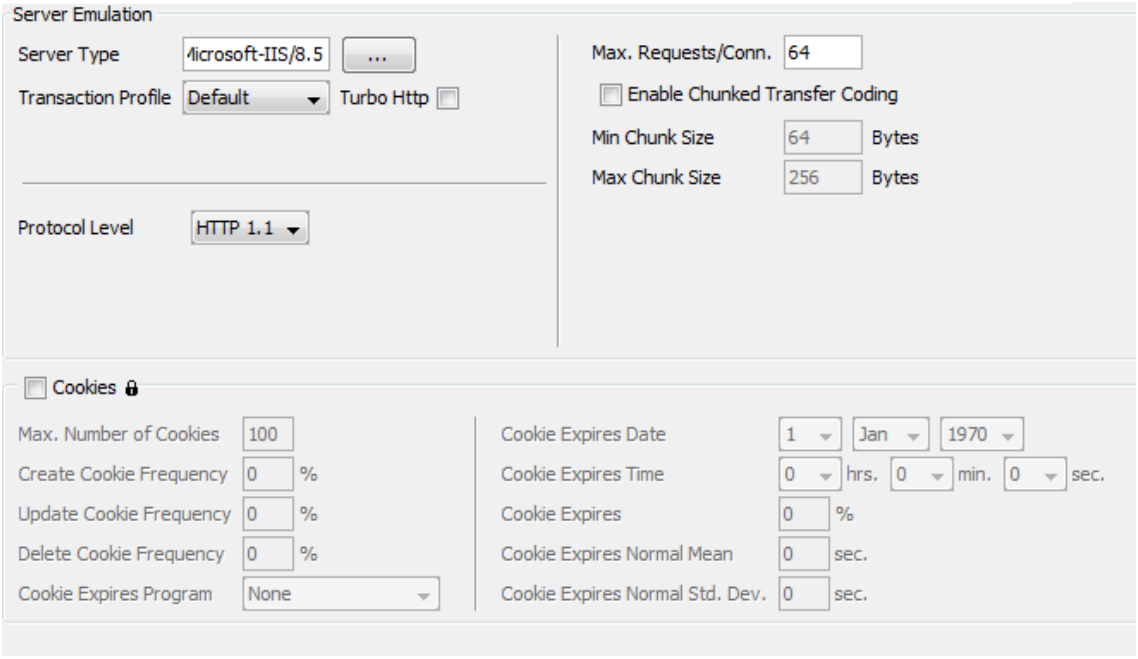
Obrázek A.41: Obecné nastavení podzáložky Server - Profiles

O něco níže nastavujeme vlastnosti spojení. V prvním řádku můžeme změnit port z výchozí hodnoty 80 na jiný. Dále můžeme aktivovat GRE (Generic Router Encapsulation), používá se, pokud chceme udělat most mezi dvěma sítěmi. Na konci řádku volíme, podle čeho se bude ukončovat spojení, ve výchozím nastavení je FIN. Ve druhém řádku můžeme nastavit hodnotu pro ToS (Type of Service). Dále nastavit ignorování žádostí protokolů ARP/NDP, na konci řádku je doplňující nastavení k nastavení ukončování spojení. Ve třetím řádku řádku můžeme upravit parametr MSS (Maximum Segment Size), kterým definujeme maximální možnou velikost příchozího segmentu TCP (bez hlavičky), pokud nic nevložíme, tak bude nastavena výchozí hodnota (pro IPv4 to je 1460 a pro IPv6 1440).



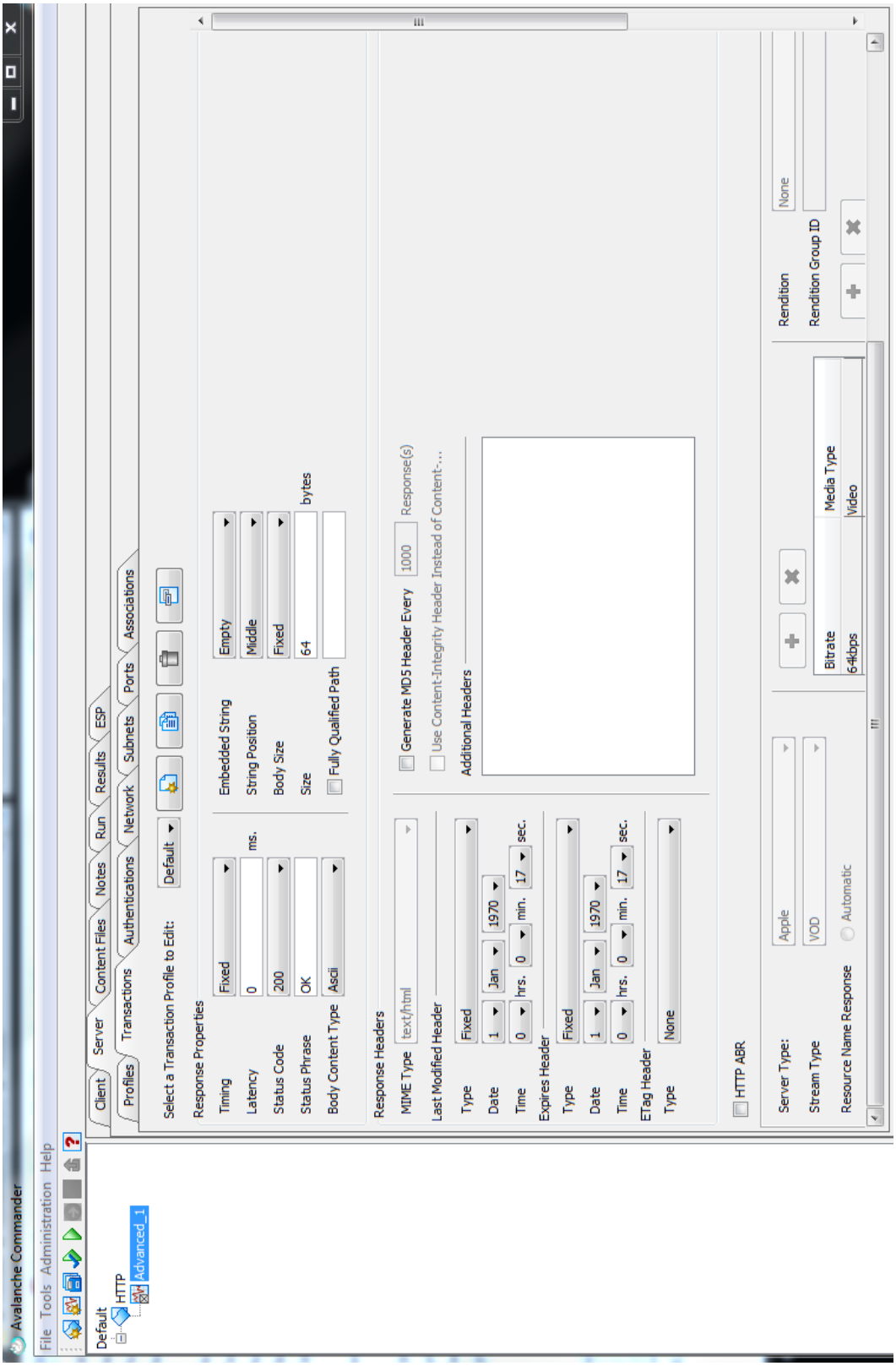
Obrázek A.42: Vlastnosti spojení

V další části měníme typ HTTP serveru, na výběr je několik verzí serveru od Microsoftu, kromě serverů od Microsoftu je možné simulovat server Apache a Nginx. Pod touto volbou je nastavení profilu transakcí, které se nastavují v další podzáložce. Dále je možnost aktivovat Turbo mód protokolu HTTP, který zvýší celkovou výkonost testů s definovanou zátěží pomocí spojení/s. O trochu níže je volba verze protokolu HTTP, na výběr je mezi 1.0, 1.1 a /2. U verze HTTP 1.1 je možné nastavit maximální počet žádostí na 1 spojení a rozsah velikostí těl odpovědí. Na závěr je možné nastavit Cookies a jejich parametry, tento parametr je však dostupný pouze s rozšiřující licencí.



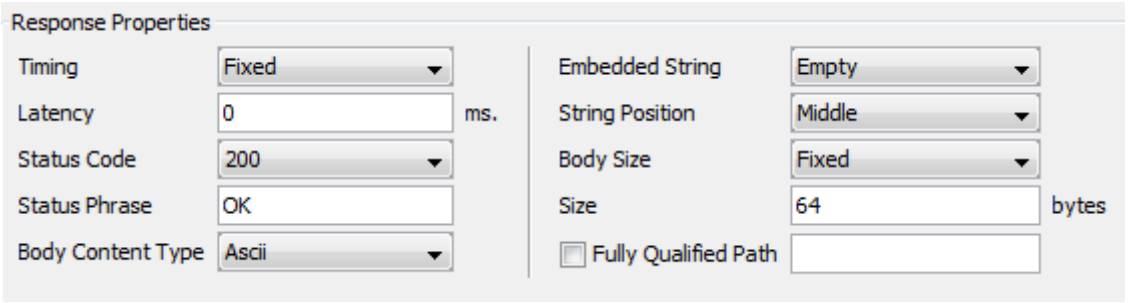
Obrázek A.43: Doplňující nastavení vybraného protokolu

Nastavení této záložky pro náš test můžeme vidět na obrázku A.40 s tím rozdílem, že bylo deaktivováno ignorování ARP/NDP žádostí.



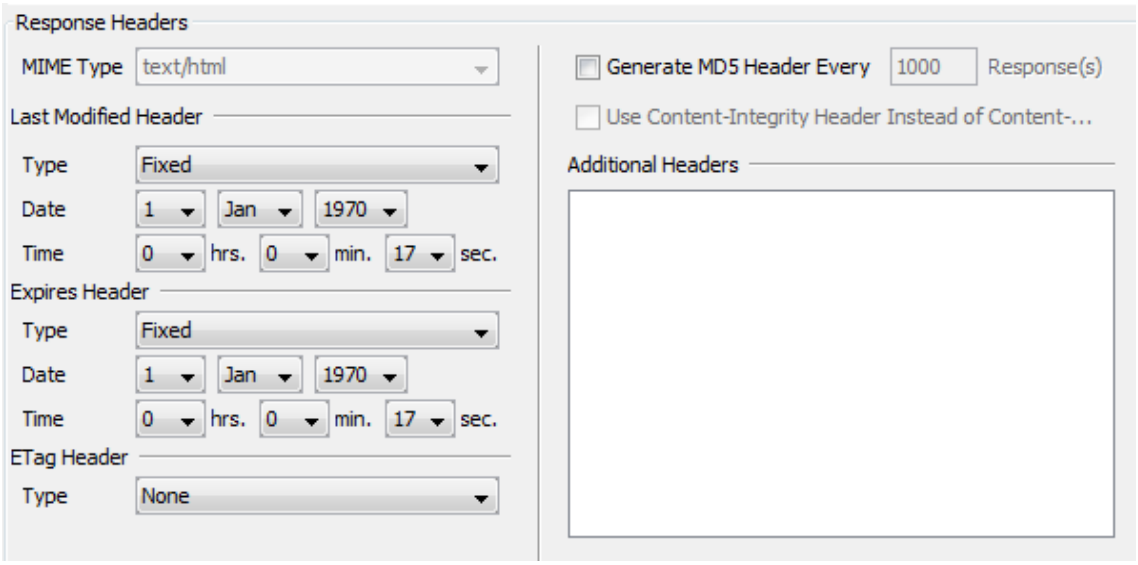
Obrázek A.44: Server - Transactions

V horní části je již dobře známé profilování. Pod ním je možné nastavit vlastnosti odpovědí. Je možné měnit například zpoždění každé odpovědi či typ obsahu těla odpovědi.



Obrázek A.45: Vlastnosti odpovědí

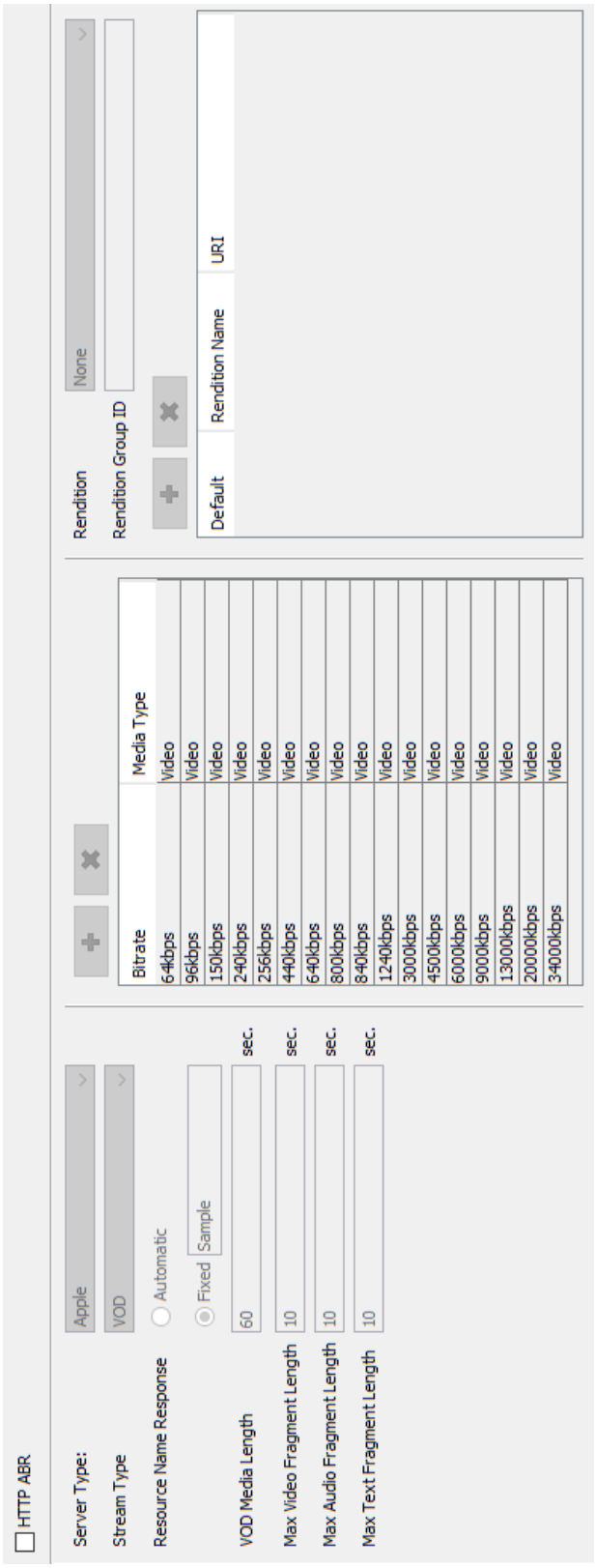
Níže je možné měnit hlavičky odpovědí, například jak často chceme generovat MD5 hlavičku. Můžeme přidat i doplňující parametry hlavičky.



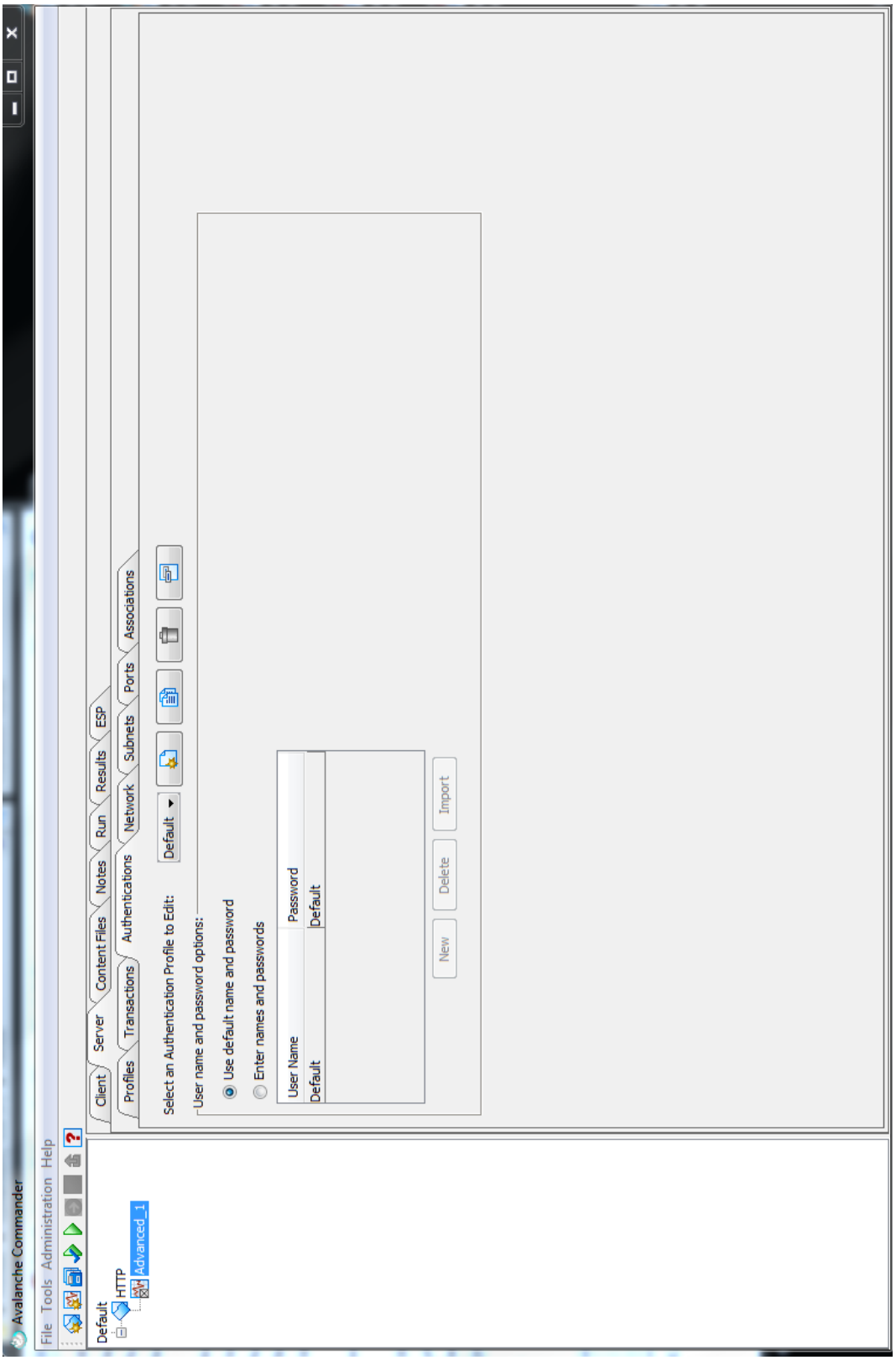
Obrázek A.46: Nastavení hlavičky odpovědí

V poslední části můžeme aktivovat HTTP ABR (adaptive streaming server). Na serveru lze nastavit typ serveru, na výběr je mezi Microsoftem, Adobe a Apple. Dále lze zvolit typ streamu, na výběr je mezi VOD (Video on Demand) nebo LIVE. Tohoto nastavení jsem ve svém měření nevyužil.

Nastavení této záložky u tohoto testu bylo ponecháno výchozí, které můžeme vidět na obrázku A.44.

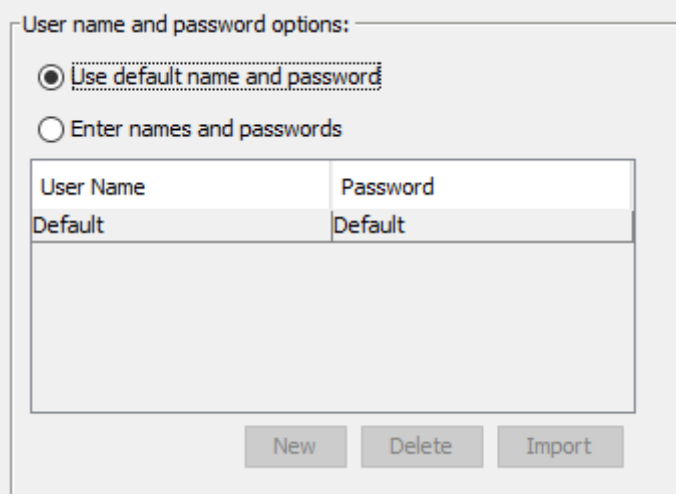


Obrázek A.47: Nastavení HTTP ABR



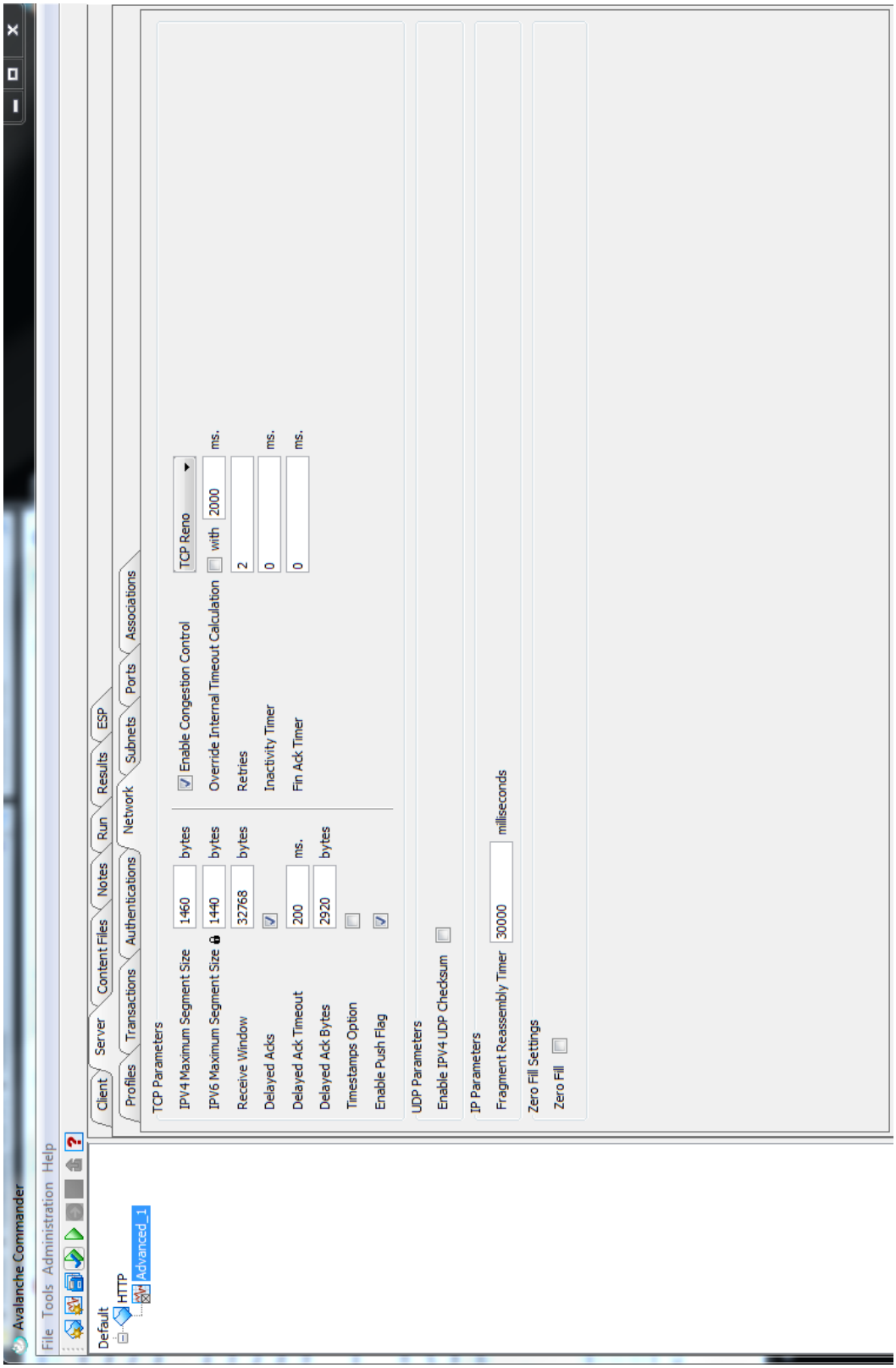
Obrázek A.48: Server - Authentications

V podzáložce je opět možnost profilování. Mimo jiné se zde primárně nastavují uživatelé, pomocí kterých se klient autentizuje na serveru. Lze zvolit buďto výchozí nastavení, nebo zakliknout Enter names and password. Poté se zpřístupní tlačítka New, Delete a Import, kterými upravujeme seznam uživatelských jmen a hesel.



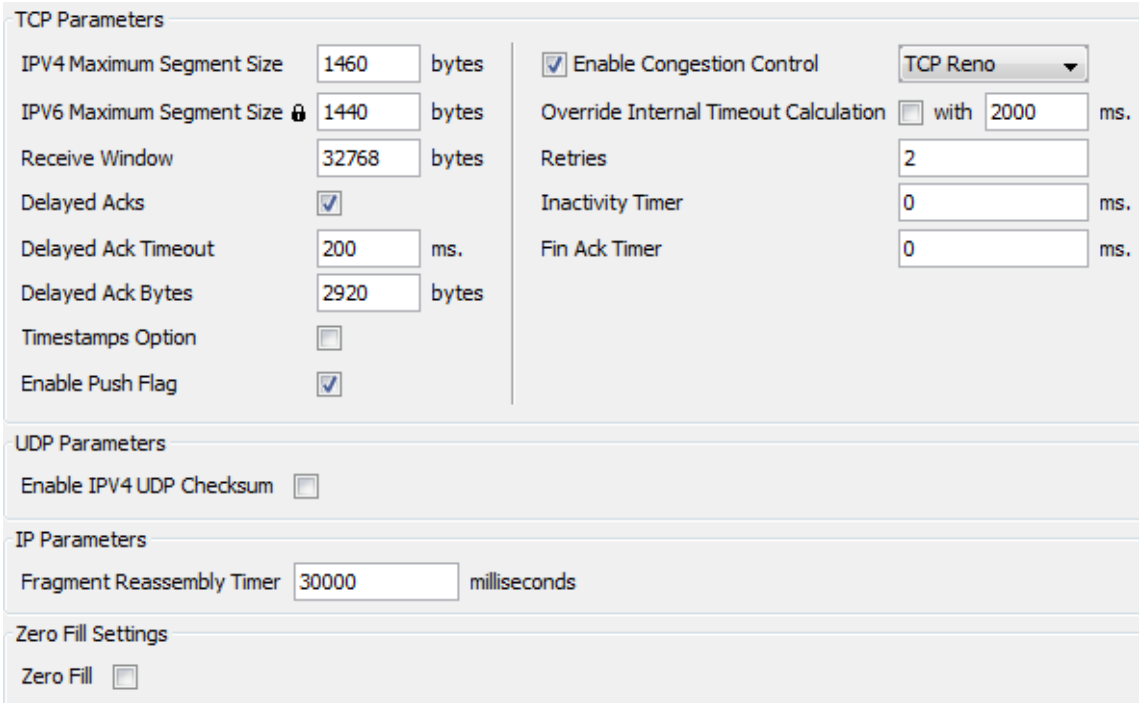
Obrázek A.49: Výchozí nastavení autentizace

U tohoto měření nebylo využito autentizace a nepřidával jsem žádné další uživatele, bylo použito výchozí nastavení, tak jak je zobrazeno na obrázku A.48.

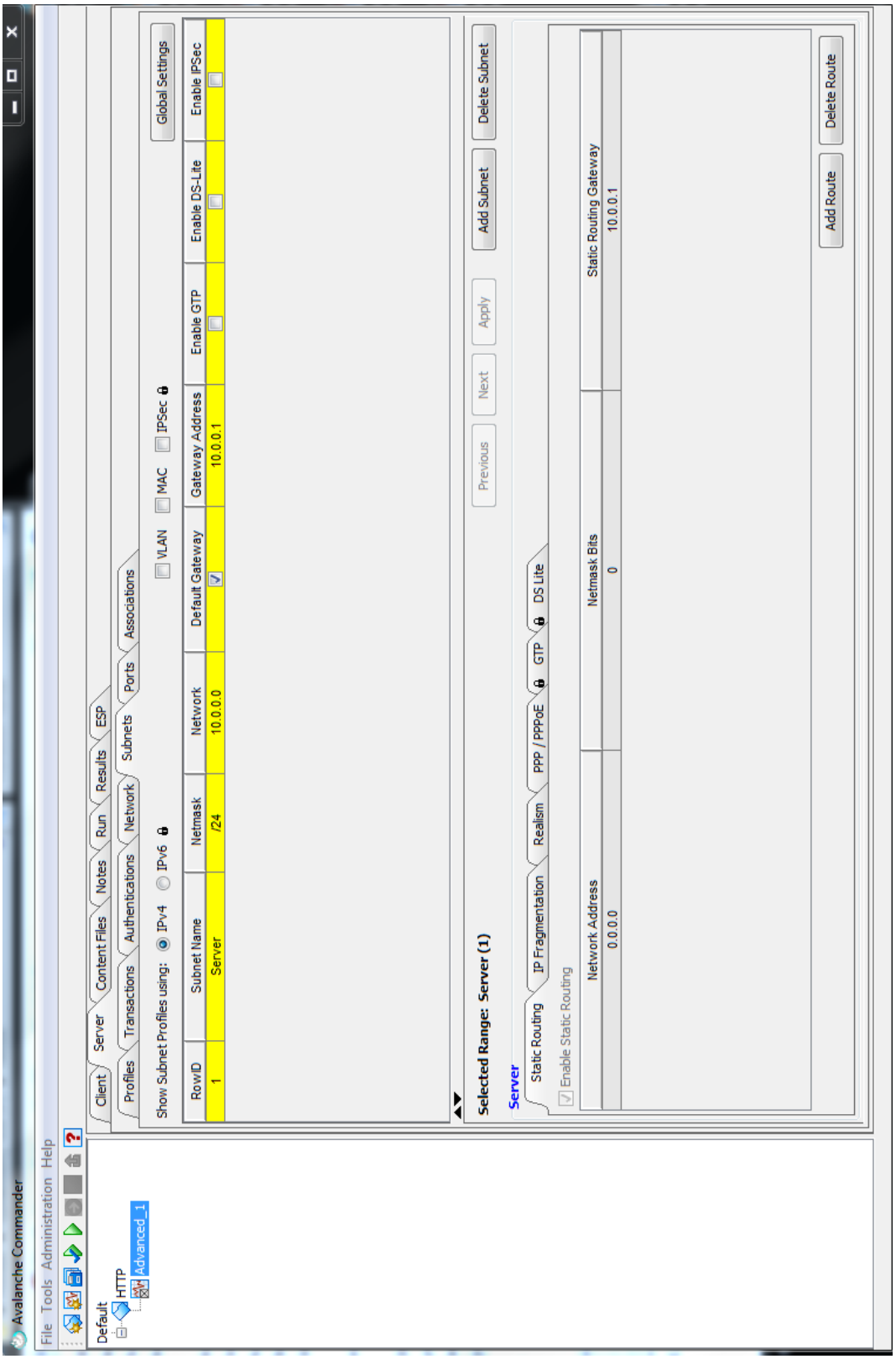


Obrázek A.50: Server - Network

Podobně jako u podzáložky Network záložky Client zde můžeme nastavit parametry týkající se transportních protokolů TCP a UDP. Taktéž je zde možnost vyplňovat nedůležitých dat nulami. Finální nastavení podzáložky Network záložky Server pro tento test můžeme vidět na obrázku A.51.



Obrázek A.51: Natavení parametrů protokolů nižších vrstev

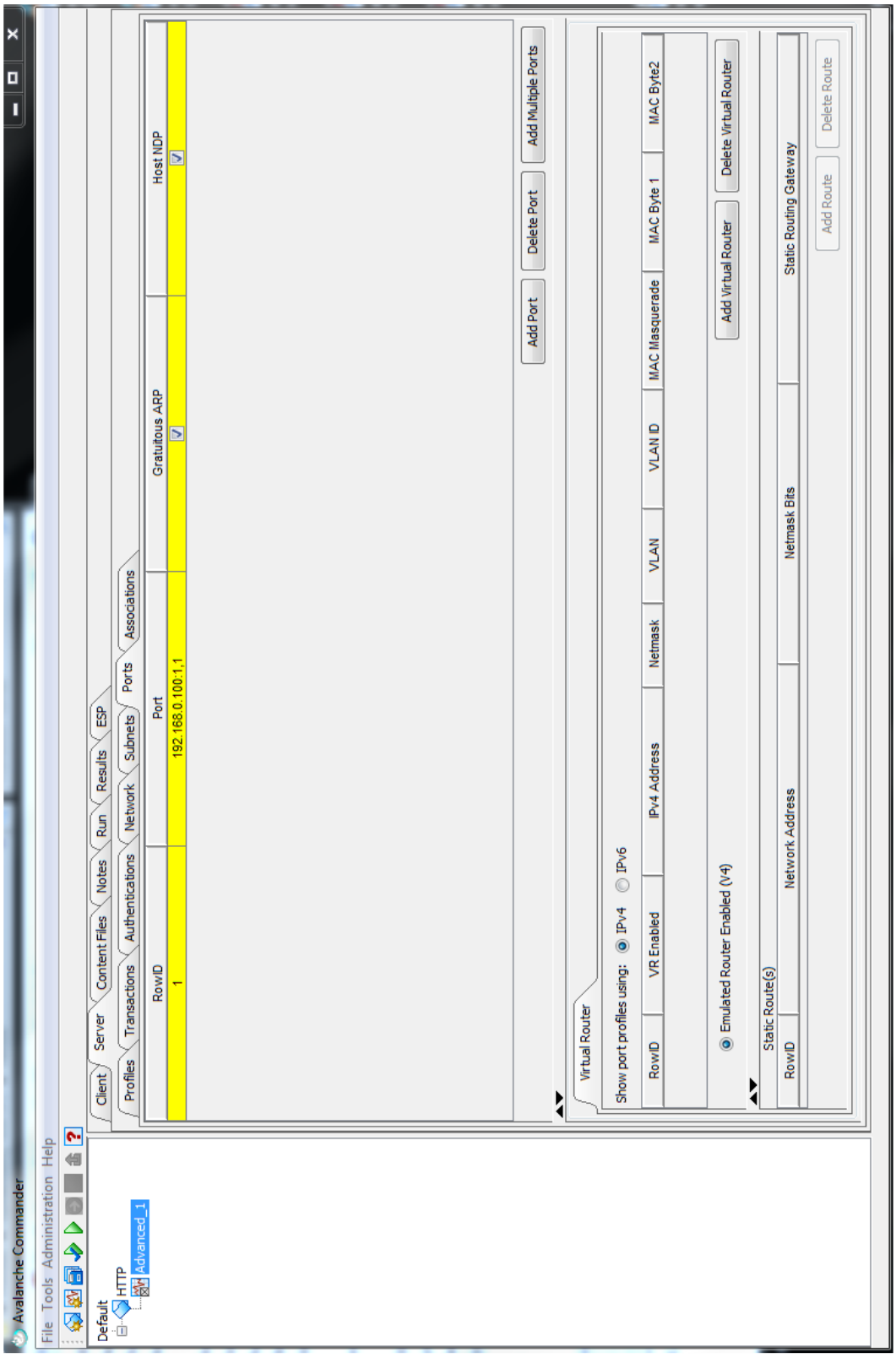


Obrázek A.52: Server - Subnets

Podzáložka Subnets záložky Server je totožná s podzáložkou Subnets záložky Client. Jediný rozdíl je v tom, že zde nenastavujeme rozsah IP adres pro klienty. IP adresa serveru se nastavuje až v podzáložce Associations. Server je v tomto testu umístěn v podsíti 10.0.0.0/24, záznam této podsítě můžeme vidět na obrázku A.53.

RowID	Subnet Name	Netmask	Network	Default Gateway	Gateway Address	Enable GTP	Enable DS-Lite	Enable IPSec
1	Server	/24	10.0.0.0	☑	10.0.0.1	☐	☐	☐

Obrázek A.53: Záznam podsítě - Server

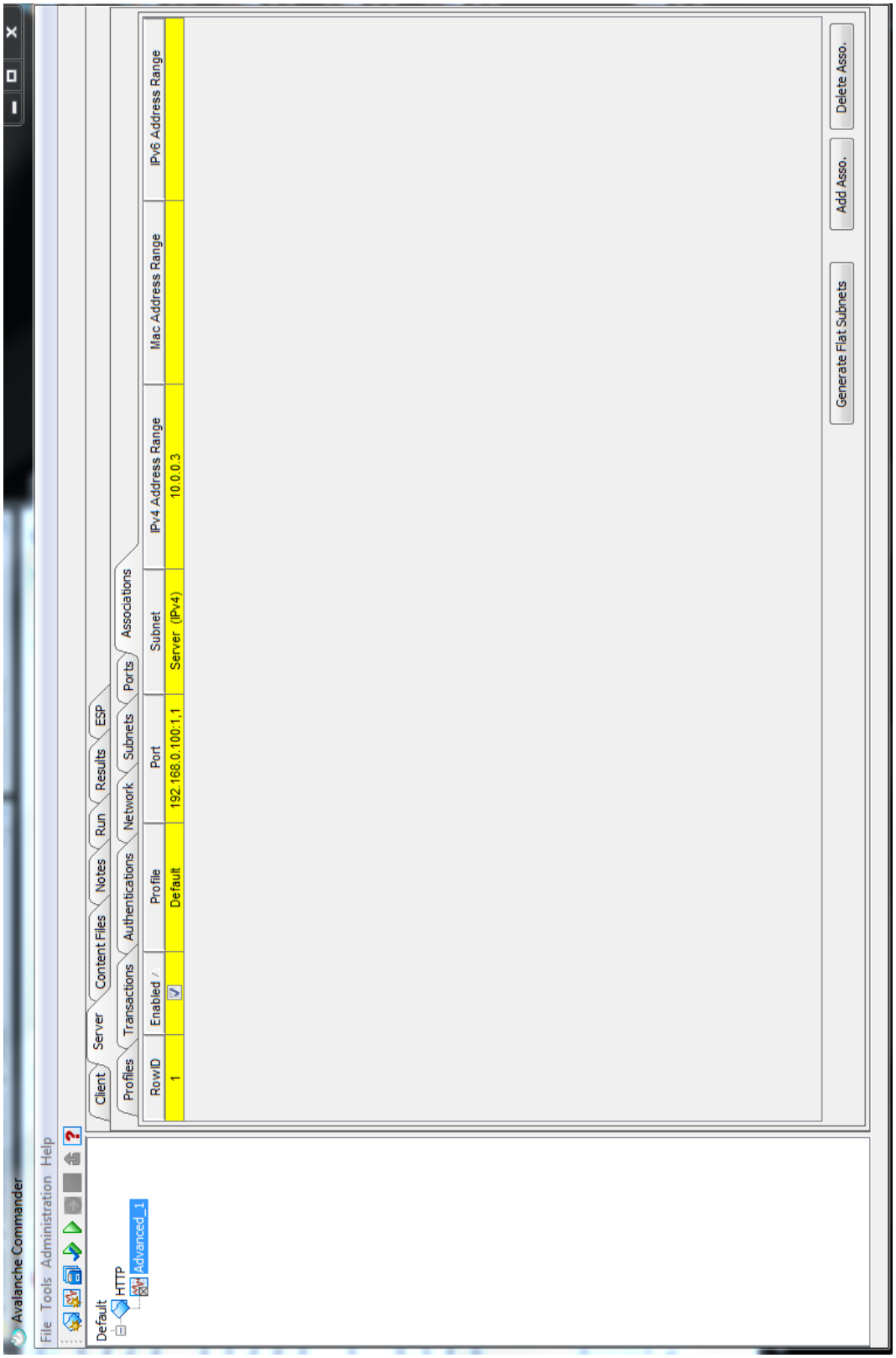


Obrázek A.54: Server - Ports

Podzáložka Ports záložky Server je opět velice blízká podzáložce Ports záložky Client. Jediný rozdíl je, že zde není umožněno generovat DDoS útoky a testovat DNS. Jelikož byl na straně klienta zvolen port 4 tak na straně serveru musíme zvolit druhý dostupný port a tím je 1. Výsledný záznam je na obrázku A.55.

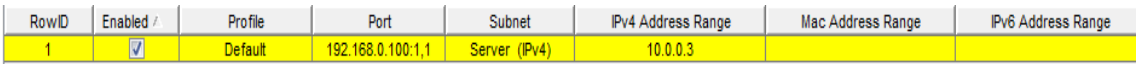
RowID	Port	Gratuitous ARP	Host NDP
1	192.168.0.100:1,1	☑	☑

Obrázek A.55: Záznam portu - Server



Obrázek A.56: Server - Associations

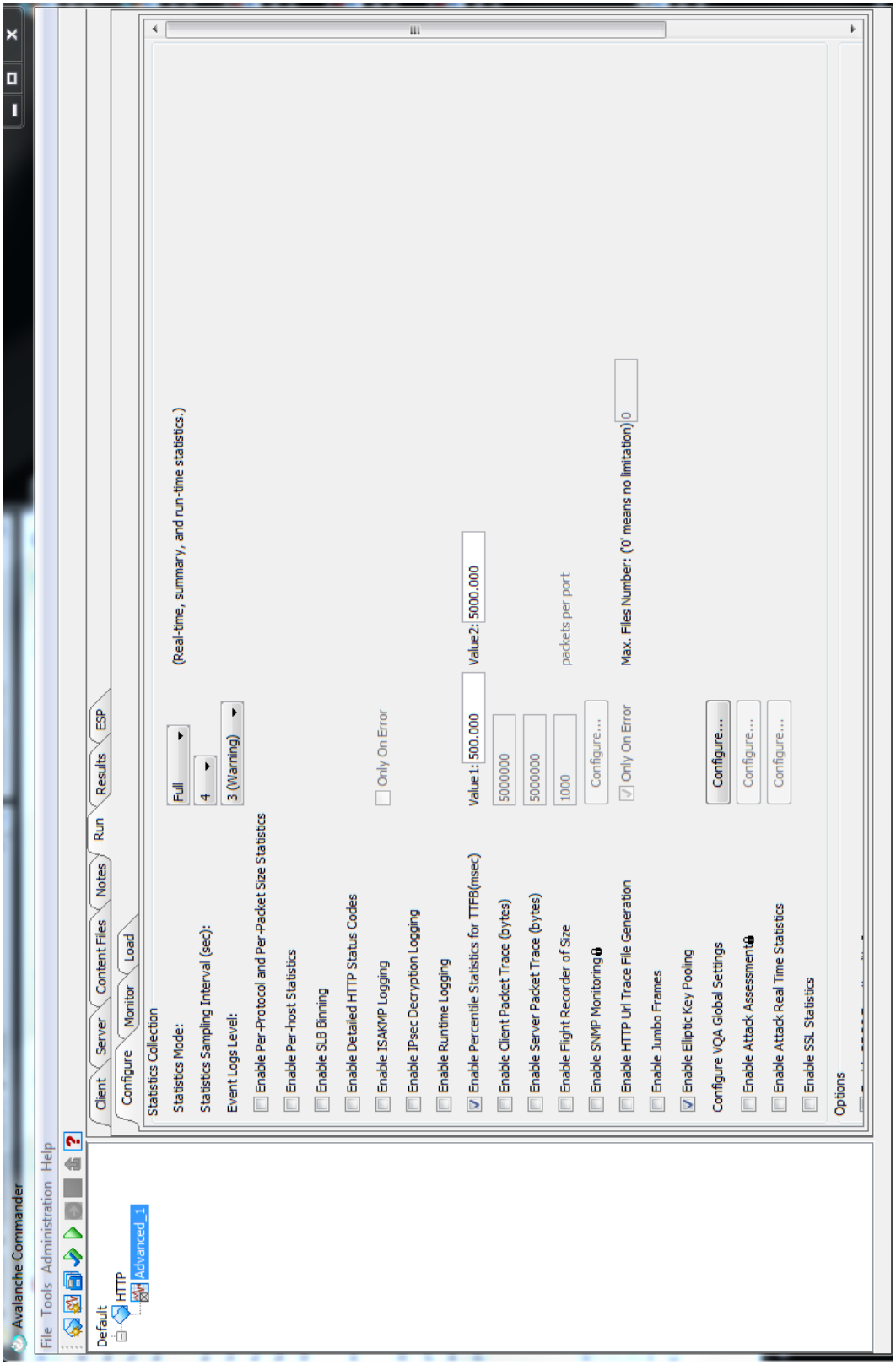
V této záložce spojujeme předem vytvořené nastavení ostatních záložek. Pokud jsme nepřidávali profily, tak stačí stisknout tlačítko Add Association a následně nastavit IP adresu serveru. Pokud jsme vytvářeli profily tak je nutné tyto profily nastavit, tak jako u podzáložky Associations záložky Client. V tomto případě byla nastavena IP adresa serveru na 10.0.0.3.



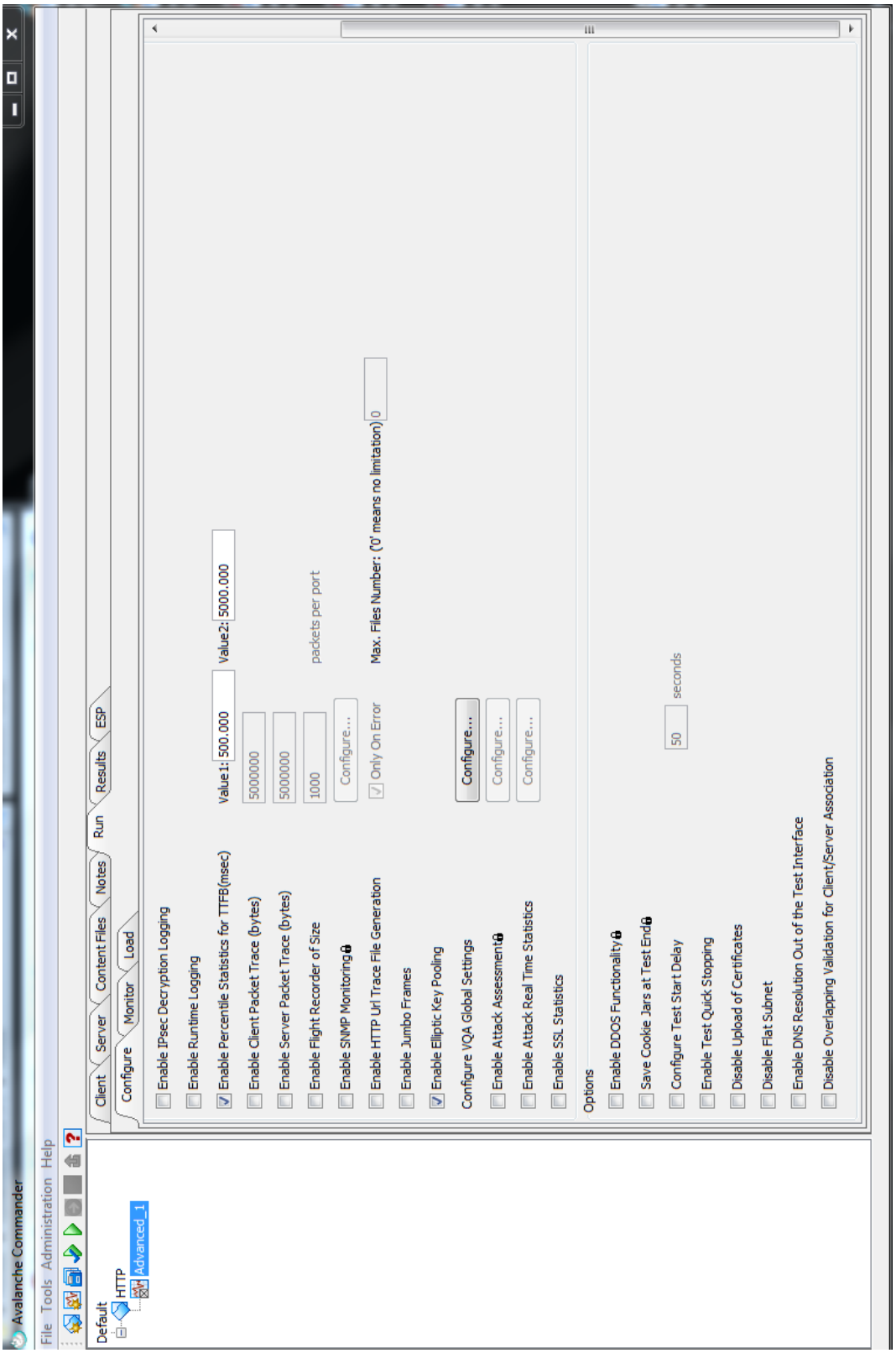
RowID	Enabled	Profile	Port	Subnet	IPv4 Address Range	Mac Address Range	IPv6 Address Range
1	☑	Default	192.168.0.100:1,1	Server (IPv4)	10.0.0.3		

Obrázek A.57: Vytvořený záznam spojených podzáložek záložky Server

Záložky Content Files a Notes nejsou nijak důležité pro spuštění testu, a tudíž se můžeme přesunout k nastavení analyzátoru.



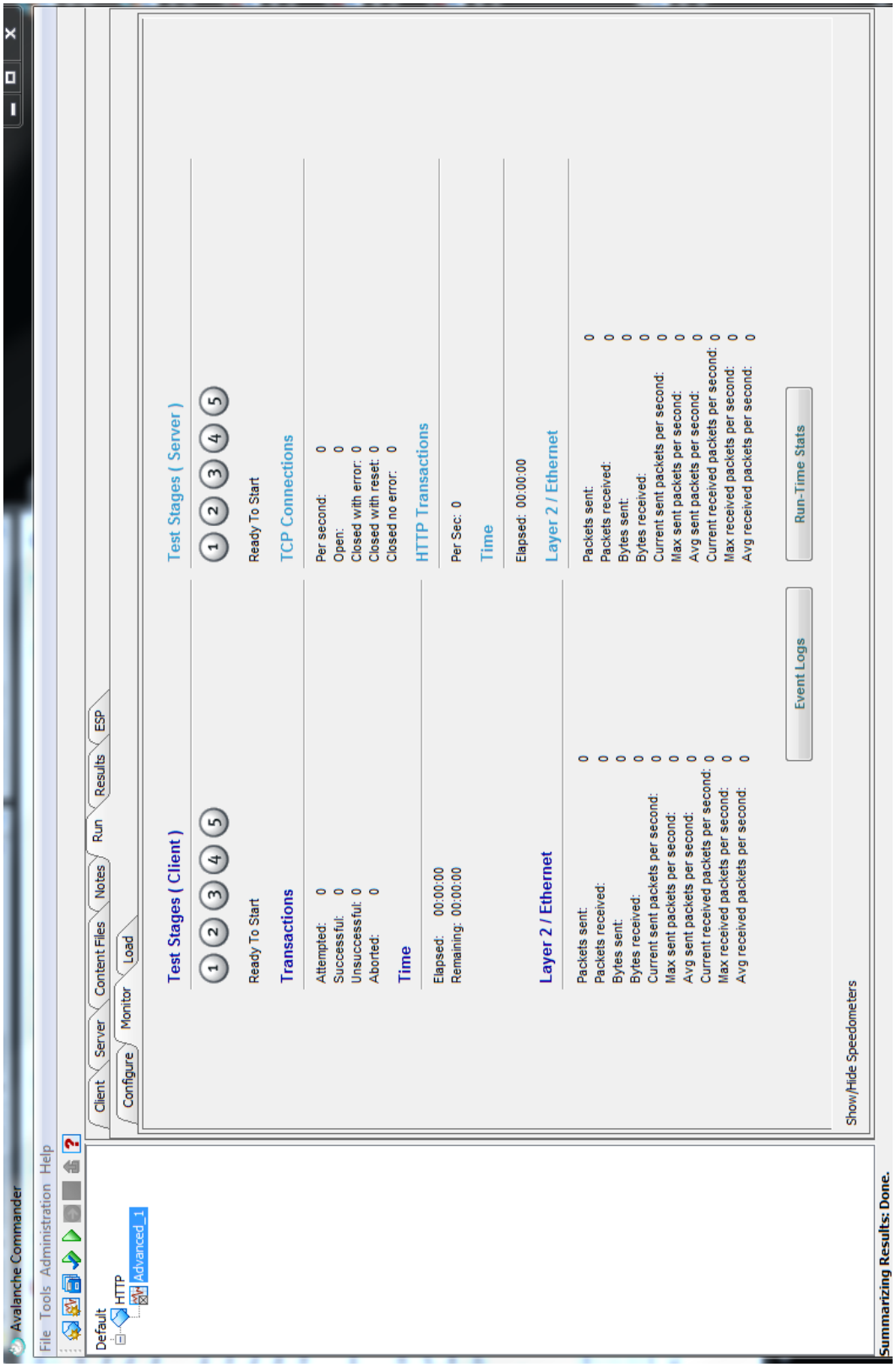
Obrázek A.58: Run - Configure 1



Obrázek A.59: Run - Configure 2

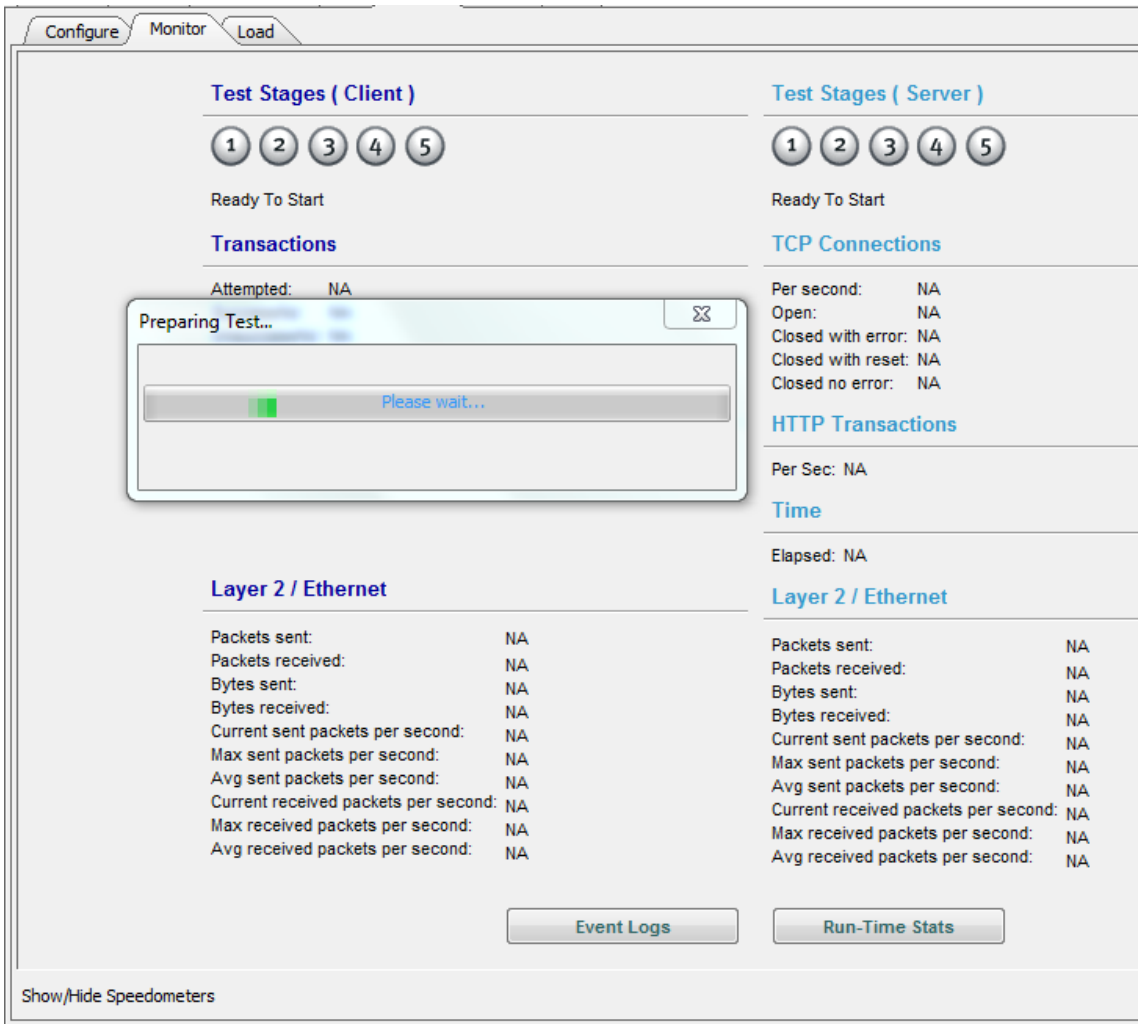
V podzáložce Configure záložky Run můžeme nastavit parametry, které chceme zachytávat během testu. Při tomto měření bylo využito výchozího nastavení podzáložky Configure a nepřidával jsem žádné další parametry k zachytávání (obrázky A.58 a A.59).

Nyní je vše připraveno k měření, a tak se můžeme přesunout do podzáložky Monitor, kde můžeme sledovat průběh testu.

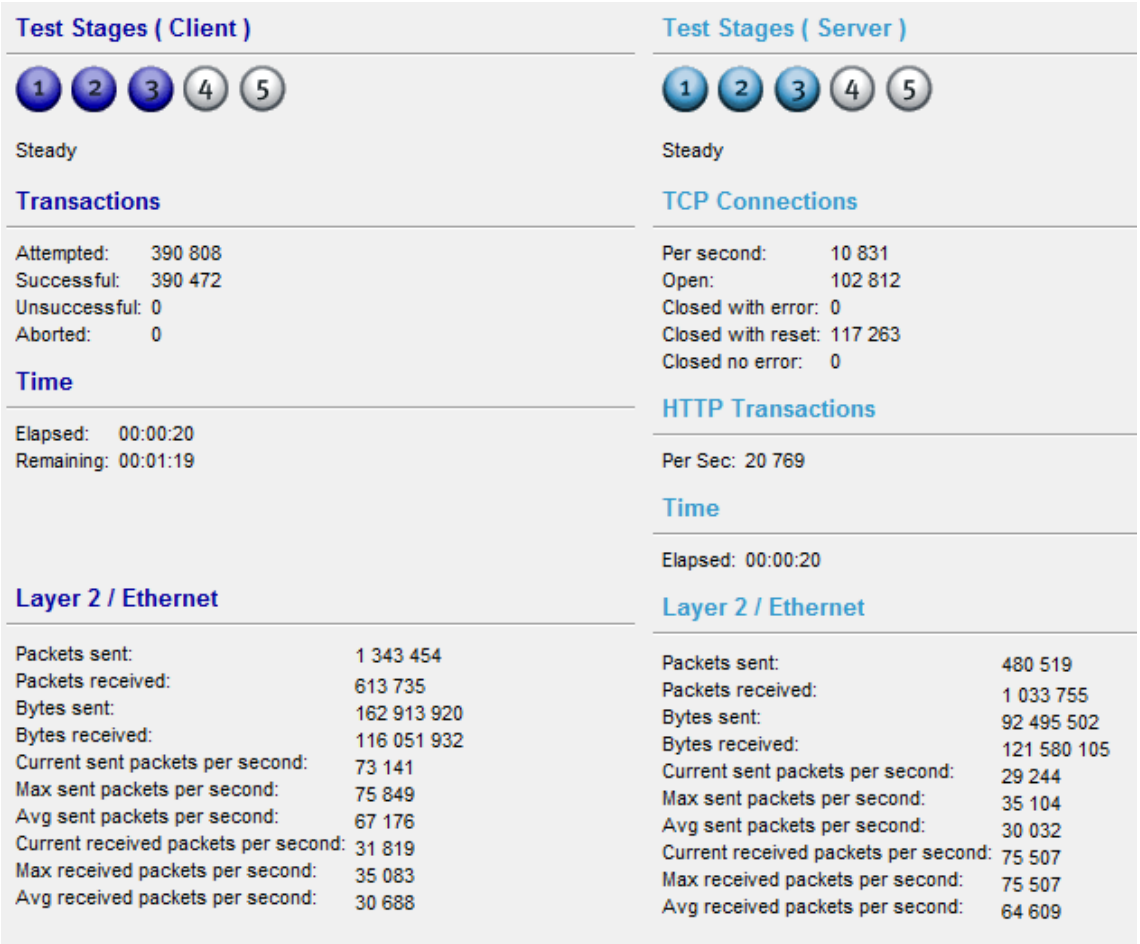


Obrázek A.60: Run - Monitor

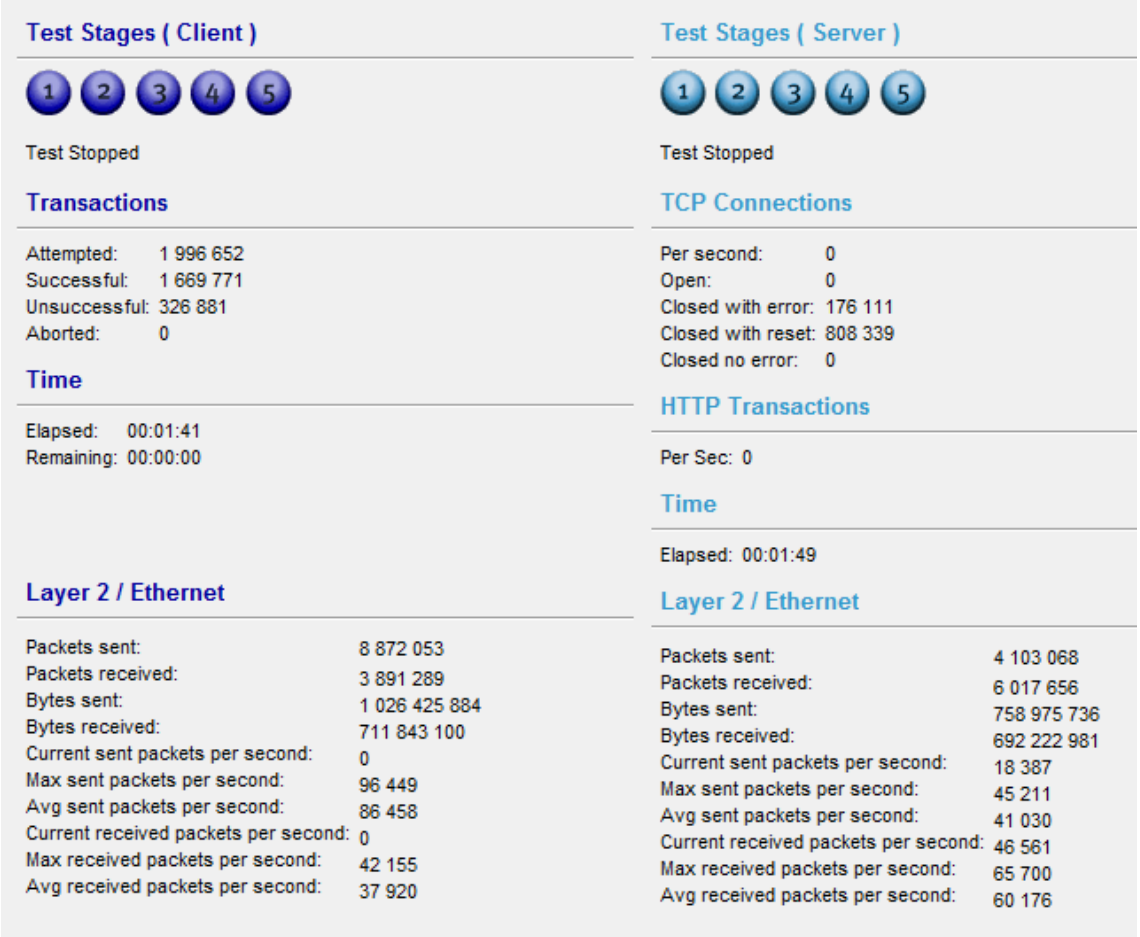
Test spustíme v horní liště pátou ikonou zleva. Ikona má tvar zeleného trojúhelníku. Následně začne příprava testu.



Obrázek A.61: Příprava testu

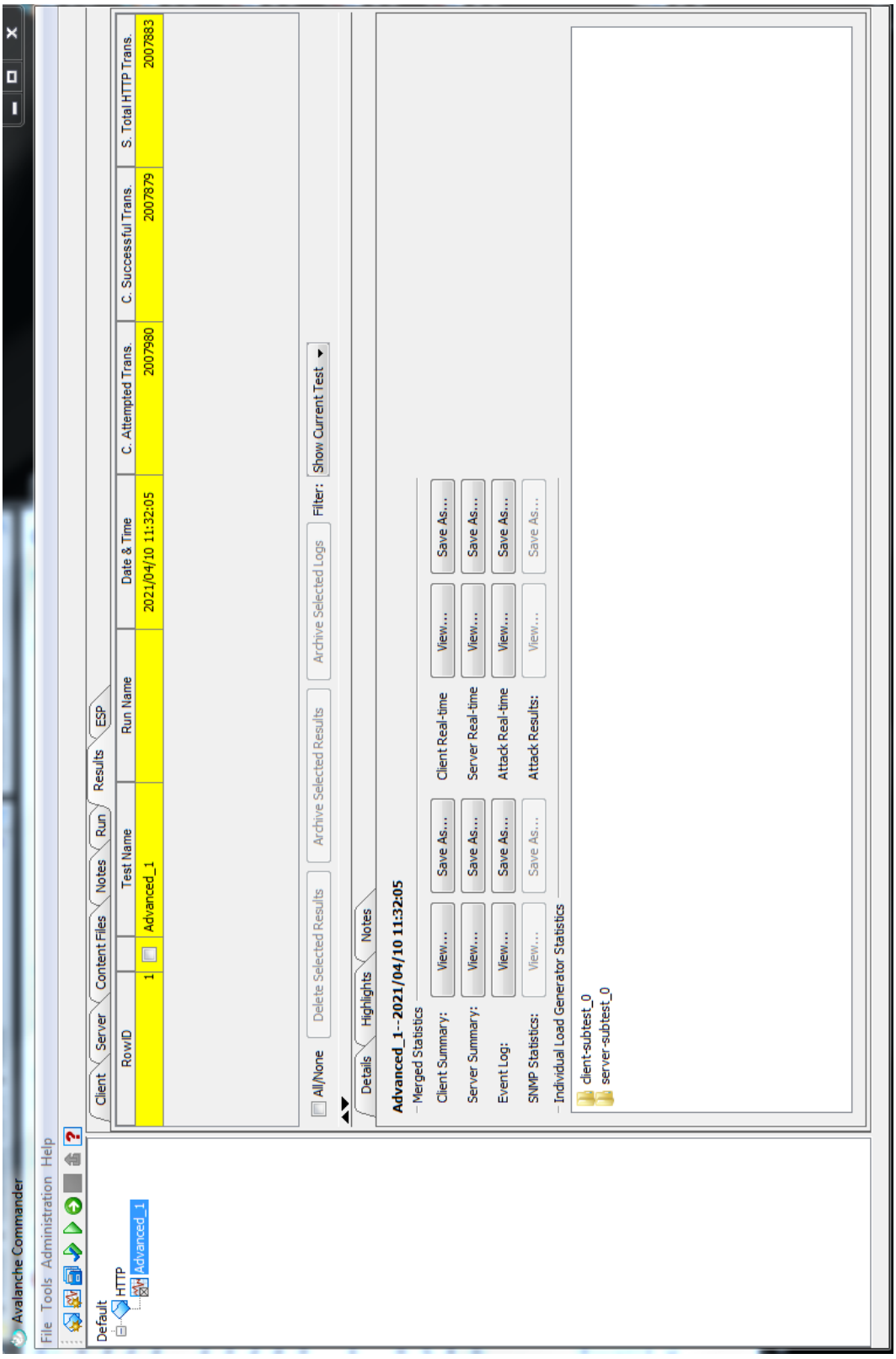


Obrázek A.62: Průběh testu



Obrázek A.63: Test dokončen

Po dokončení testu se vytvoří záznam v záložce Results. Statistiky můžeme ihned zobrazit tlačítkem View… nebo můžeme samotný test uložit na disk pro pozdější použití tlačítkem Archive Selected Results.

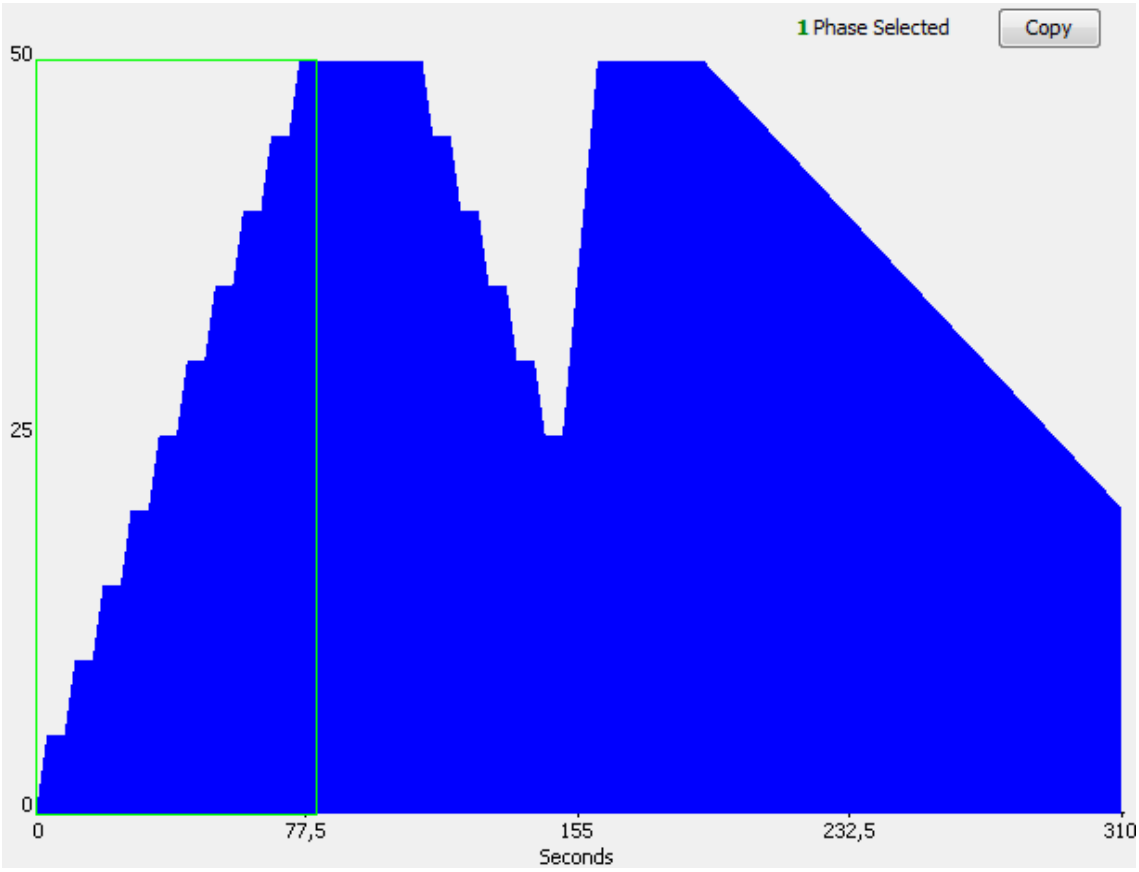


Obrázek A.64: Results

B Nastavení pro měření parametrů protokolu FTP

Postup měření je podobný tomu, který je uveden v kompletním návodu (příloha A). Z toho důvodu zde budu popisovat pouze rozdílné části měření.

V podzáložce Loads záložky Client jsem v základním nastavení specifikoval zátěž pomocí počtu Spojení (Connections). Dále jsem v editorů fází nastavil tvar provozu do podoby, kterou můžeme vidět na obrázku B.1.



Obrázek B.1: Tvar generované zátěže (FTP)

Druhou upravovanou podzáložkou záložky Client byla Actions. Zde jsem generoval metodu GET s použitím autentizace. Tvar generované metody můžeme vidět na obrázku B.2.

V rámci protokolu FTP můžeme generovat dvě různé akce (GET a PUT). Základní syntaxi příkazu GET můžeme vidět níže, 1b za IP adresou serveru značí velikost souboru, který budeme stahovat. Minimum je 1 Byte (1b) a maximum GigaByte (1g). Pokud používáme reálný server tak za lomítko vkládáme cestu a jméno požadovaného souboru.

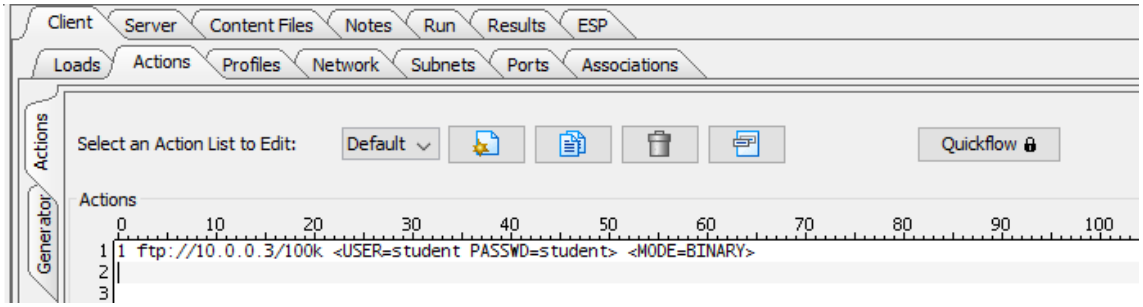
```
1 ftp://10.0.0.3/1b
```

Obecný zápis pro nahrání/stažení souboru test.txt můžeme vidět na následujících řádcích.

```
1 ftp://10.0.0.3 <PUT = test.txt>
```

```
1 ftp://10.0.0.3 <GET = test.txt>
```

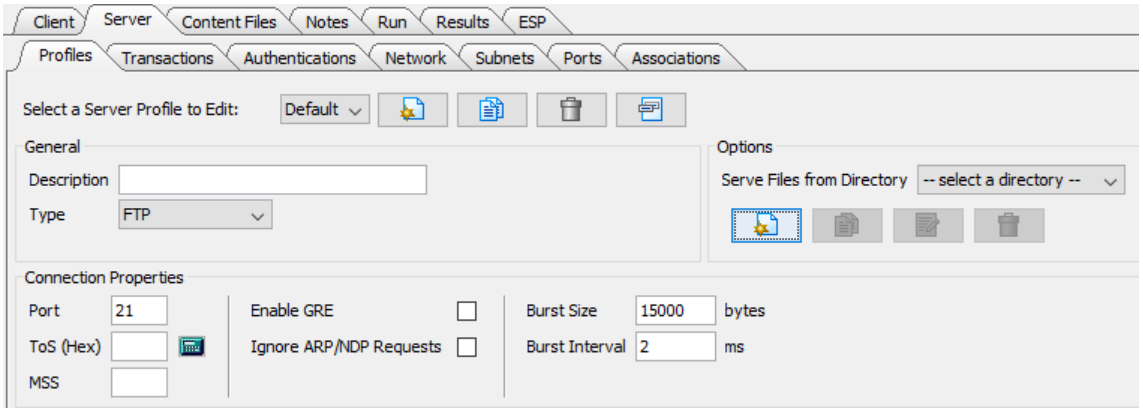
Podobně jako protokol HTTP, tak i protokol FTP má k dispozici doplňující parametry jako je například autentizace nebo mód přenosu, které můžeme vidět na obrázku B.2. USER=student nám říká, že se budeme přihlašovat s uživatelským jménem student a pomocí hesla PASSWD=student se u serveru autentizujeme. Kromě binárního módu přenosu můžeme zvolit i alternativní mód ASCII.



Obrázek B.2: Generovaná akce (FTP)

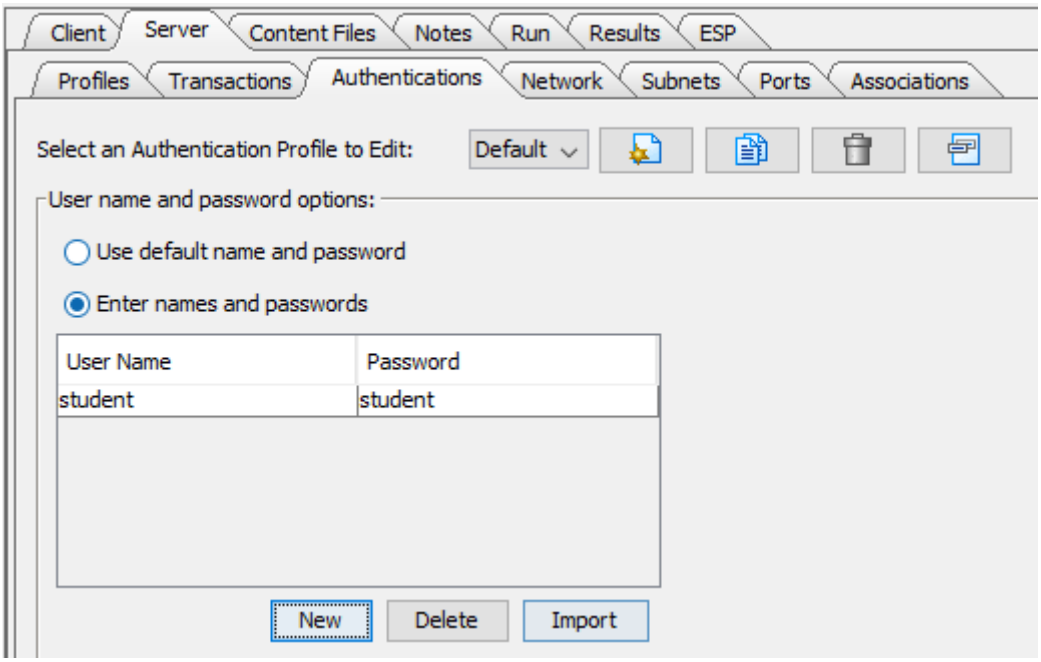
V podzáložce Profiles záložky Client žádné z dostupných tlačítek neovlivňuje měření protokolu FTP. Proto zde bylo ponecháno výchozí nastavení. Podzáložky Network, Subnets, Ports a Associations záložky Client měly totožnou formu jako u testování protokolu HTTP.

V podzáložce Profiles záložky Server jsem v obecném nastavení změnil typ serveru na FTP. V nastavení spojení serveru jsem vypnul ignorování žádostí protokolů ARP/NDP. Mimo jiné lze nastavit Burst size, kterým říkáme, jak velká má být velká odpověď serveru. Burst interval určuje čekací dobu mezi odpověďmi. Mimo jiné je zde nastavení Options, kde můžeme přidat složku se soubory, které budou na simulovaném serveru.



Obrázek B.3: Nastavení Server - Profiles (FTP)

Další upravenou podzáložkou byla Authentiactions. Zde jsem přidal do seznamu uživatelů záznam, kde uživatelské jméno bylo student a heslo také student. Tento záznam můžeme vidět na obrázku B.4.



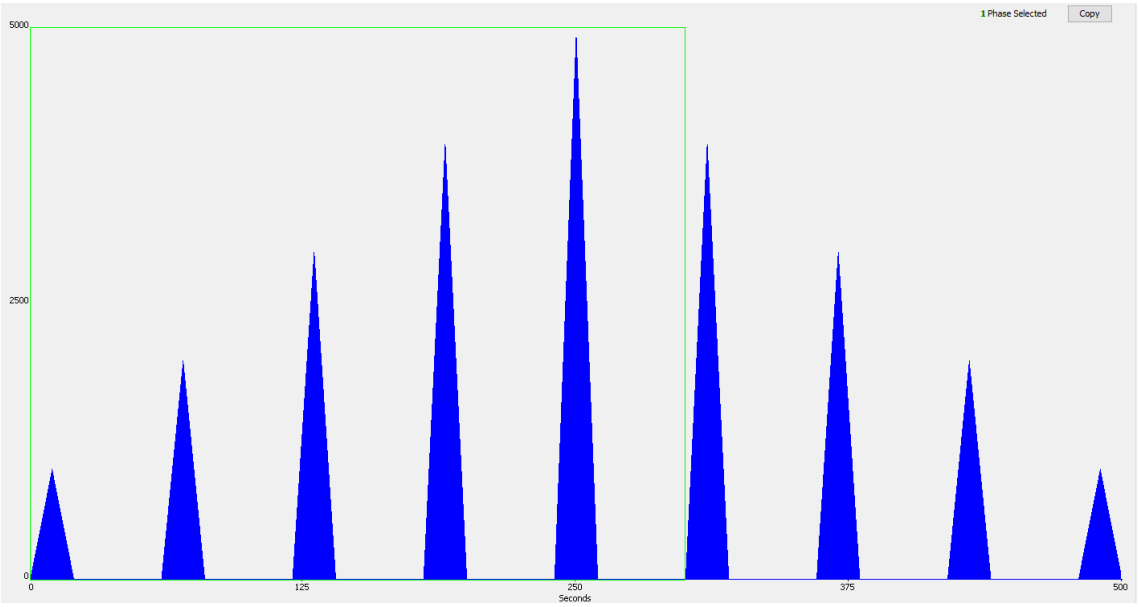
Obrázek B.4: Záznam v podzálože Authentications (FTP)

Zbylé nastavení bylo totožné s tím u měření protokolu HTTP z přílohy A.

C Nastavení pro měření parametrů protokolu SMTP

Postup měření je podobný tomu, který je uveden v kompletním návodu (příloha A). Z toho důvodu zde budu popisovat pouze rozdílné části měření.

V podzáložce Loads záložky Client jsem v základním nastavení specifikoval zátěž pomocí počtu Simulovaných uživatelů (SimUsers). Dále jsem v editorů fází nastavil tvar provozu do podoby, kterou můžeme vidět na obrázku C.1.



Obrázek C.1: Tvar generované zátěže (SMTP)

Druhou upravovanou podzáložkou záložky Client byla Actions. Zde je možné generovat pouze jednu akci a tím je posílání e-mailů. Tvar generované akce můžeme vidět na obrázku C.2. U samotné akce můžeme měnit různé parametry. V základním tvaru je nutné zadat e-mailovou adresu příjemce, odesílatele a velikost samotné zprávy. Mezi doplňující parametr patří například SUBJECT, kterým definujeme předmět e-mailu. K definování obsahu zprávy slouží parametr DATA, zde můžeme vložit libovolnou zprávu, nebo zprávu definovat pomocí velikosti, tak jako to je na obrázku C.2. Níže můžeme vidět další možné způsoby zápisu. V prvním příkladě vkládáme libovolný text zprávy, ve druhém příkladě generujeme zprávu o náhodné velikosti z rozsahu 0 až 1000 Bytů. Ve třetím příkladě generujeme zprávu o fixní velikosti 300 Bytů s tím, že je k mailu přiložen pdf soubor s názvem PDFDoc.

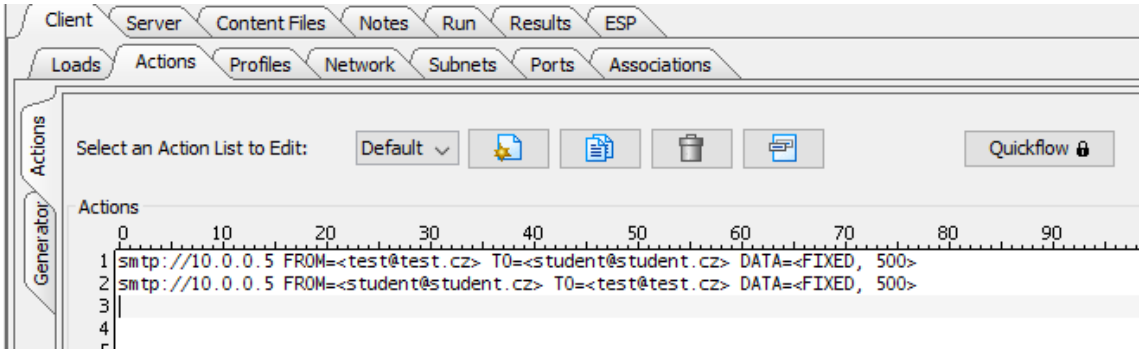
```
smtp://10.0.0.3 FROM=<test@test.cz> TO=<student@student.cz>
SUBJECT=<"Předmět"> DATA=<Text zprávy.>
```

```
smtp://10.0.0.3 FROM=<test@test.cz> TO=<student@student.cz>
SUBJECT=<"Předmět"> DATA=<RANDOM=UNIFORM 0 1000>
```

```
smtp://10.0.0.3 FROM=<test@test.cz> TO=<student@student.cz>
SUBJECT=<"Předmět"> DATA=<FIXED, 300> SMTP_ATTACH_FILE=<"PDFDoc"
"application/pdf">
```

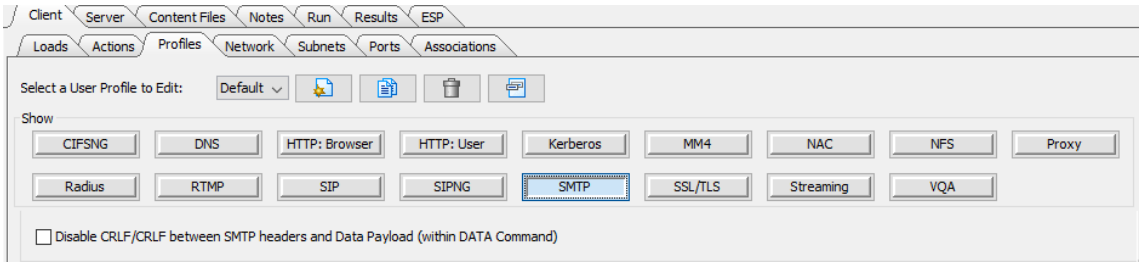
SMTP protokol umožňuje generování jeho vylepšené verze ESMTP (Enhanced SMTP). Tato verze se liší v tom, že je šifrovaná a musí přiložit k příkazu autentizaci. Syntaxi takové akce můžeme vidět níže. USER a PASSWD známe již z měření FTP (příloha B), AUTH_TYPE nám říká o jaký typ autentizace se jedná, kromě "login" můžeme nastavit plain, cram-md5 nebo digest-md5.

```
esmtp://10.0.0.3 USER=student PASSWD=student AUTH_TYPE=login
FROM=<test@test.cz> TO=<student@student.cz> DATA=<FIXED,100>
```



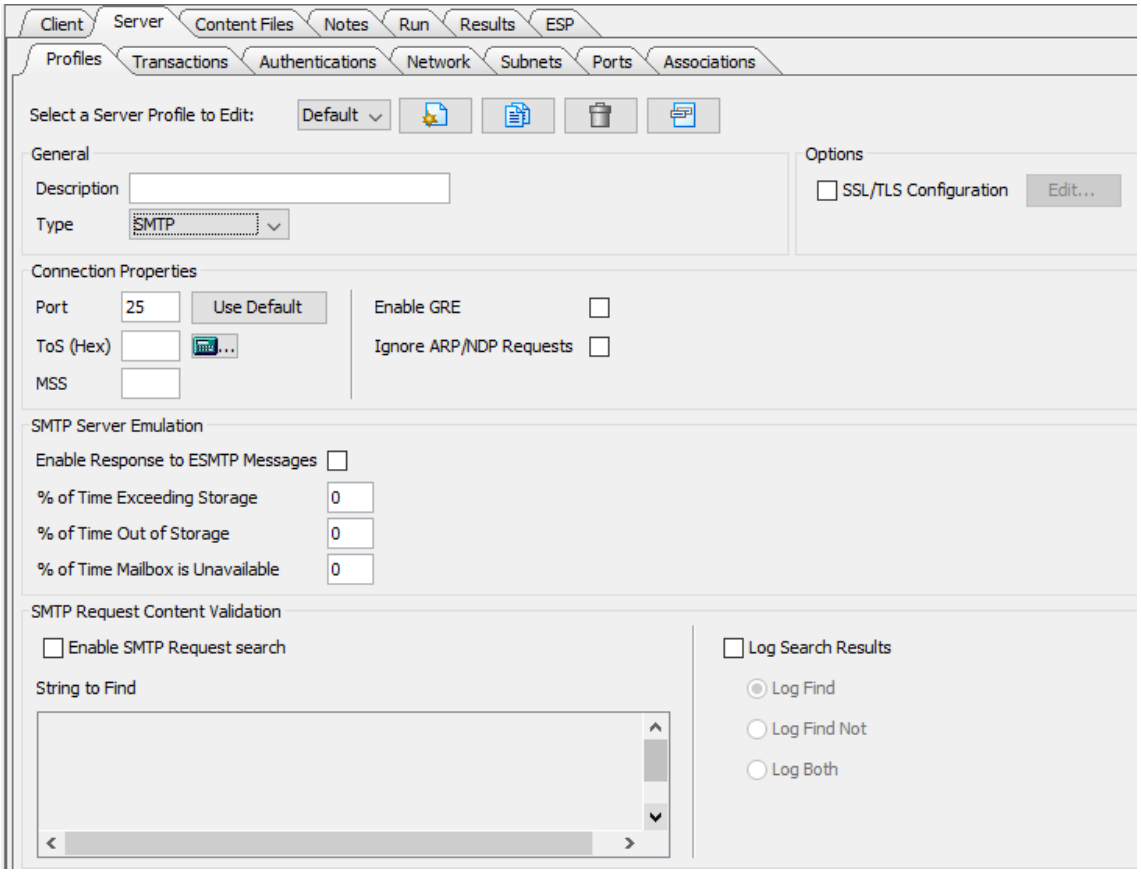
Obrázek C.2: Generovaná akce (SMTP)

V podzáložce Profiles záložky Client jsou pro SMTP dvě tlačítka, prvním z nich je SMTP, kde můžeme nastavit pouze jeden parametr a tím je deaktivování vkládání prázdného řádku mezi SMTP hlavičku a tělo zprávy. Nastavení pro tento test můžeme vidět na obrázku C.3. Kromě tlačítka SMTP můžeme nastavit i tlačítko SSL/TLS, ale to se týká pouze verze ESMTP. Zbylé nastavení záložky Client je totožné s předchozími testy z příloh A a B.



Obrázek C.3: Client - Profiles (SMTP)

V podzáložce Profiles záložky Server jsem v obecném nastavení změnil typ serveru na SMTP. V nastavení spojení serveru jsem vypnul ignorování žádostí protokolů ARP/NDP. Zbylé nastavení lze vidět na obrázku C.4. Pokud používáme ESMTP tak můžeme otevřít konfiguraci SSL/TLS. Dále můžeme nastavit, aby server odpovídal na ESMTP zprávy. Nastavit kolik procent času z testu bude e-mailová schránka nedostupná, kolik procent času z testu bude schránka plná, kolik procent času bude e-mailová schránka přesahovat uložiště. Taktéž můžeme nastavit vyhledávání testu v zprávě.



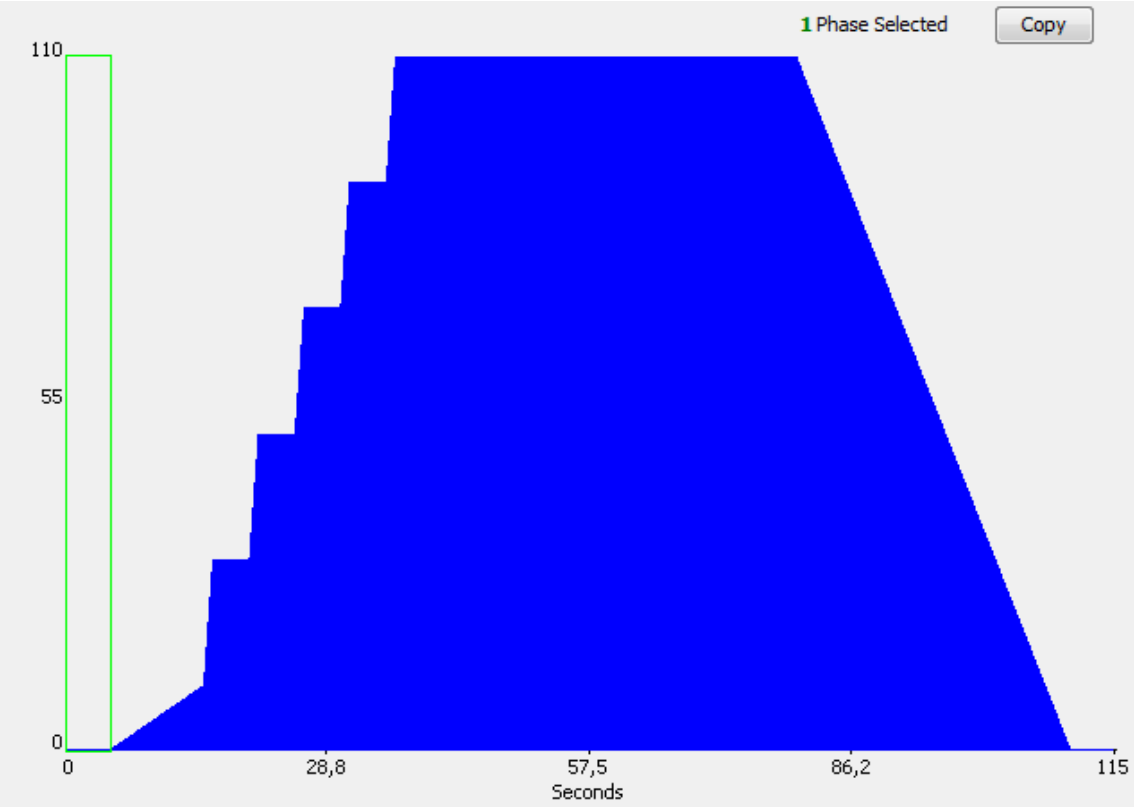
Obrázek C.4: Nastavení Server - Profiles (SMTP)

Zbylé nastavení bylo totožné s tím u měření protokolu HTTP z přílohy A.

D Nastavení pro měření několika aplikačních protokolů najednou

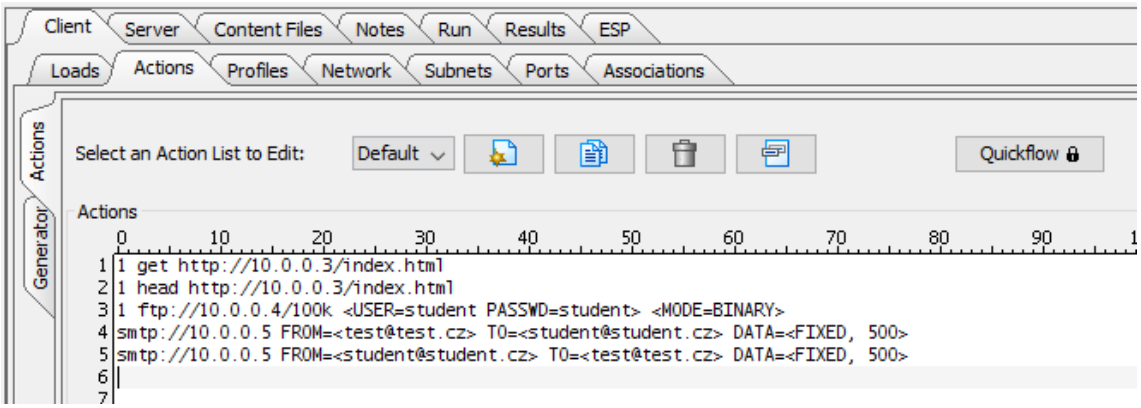
Postup měření je podobný tomu, který je uveden v kompletním návodu (příloha A). Z toho důvodu zde budu popisovat pouze rozdílné části měření.

V podzáložce Loads záložky Client jsem v základním nastavení specifikoval zátěž pomocí počtu Spojení (Connections). Dále jsem v editorů fází nastavil tvar provozu do podoby, kterou můžeme vidět na obrázku D.1.



Obrázek D.1: Tvar generované zátěže (několik aplikačních protokolů)

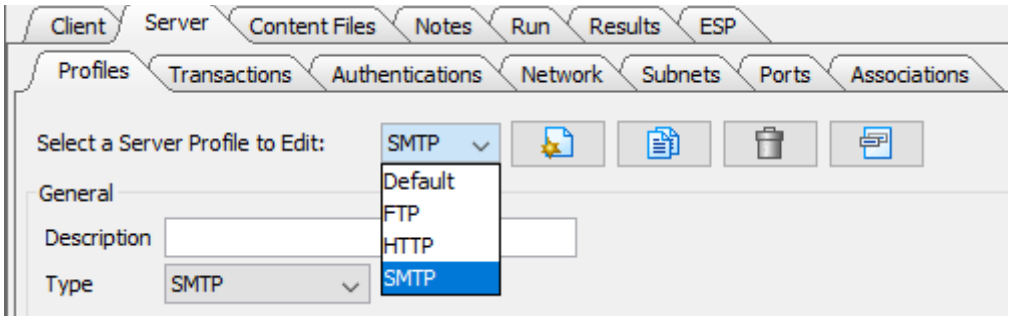
Druhou upravovanou podzáložkou záložky Client byla Actions. Zde jsem generoval akce ze všech předchozích příloh (A, B, C). Generované akce můžeme vidět na obrázku D.2.



Obrázek D.2: Generovaná akce (několik aplikačních protokolů)

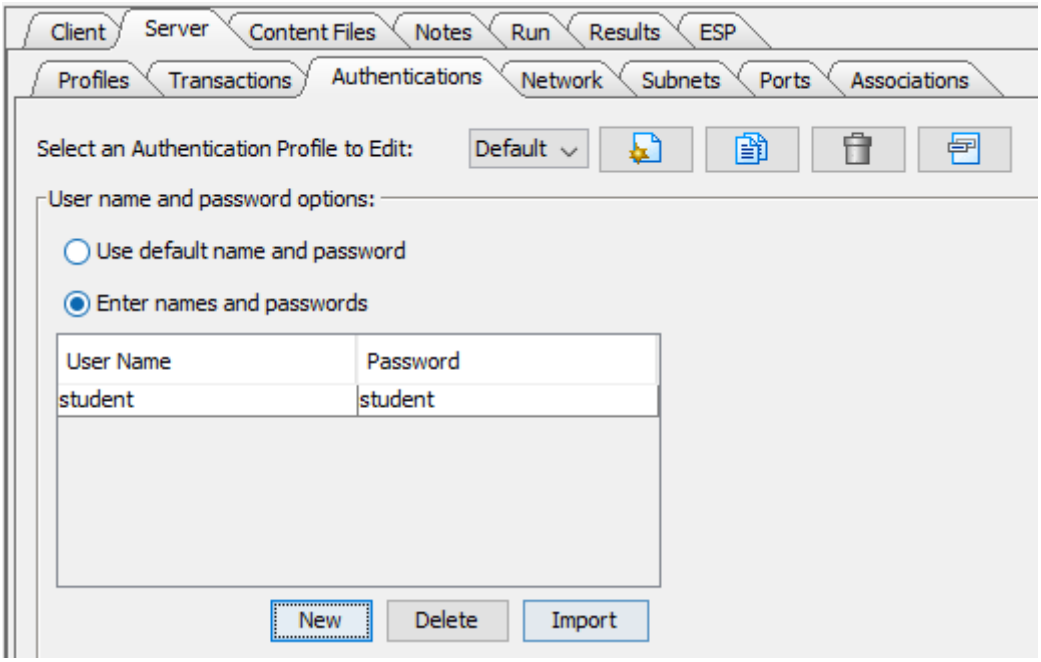
V podzáložce Profiles záložky Client jsem nastavil tlačítka HTTP:Browser a HTTP:User stejně jako v příloze A. Tlačítko SMTP bylo nastaveno stejně jako v příloze C. Ostatní podzáložky záložky Client byly totožné s předchozími testy z příloh A, B nebo C.

V podzáložce Profiles záložky Server jsem vytvořil 3 profily. Jeden pro HTTP, druhý pro FTP a třetí pro SMTP. U každého profilu jsem provedl nastavení podle předchozích testů z příloh A, B, C podle toho, o jaký typ serveru se jedná.



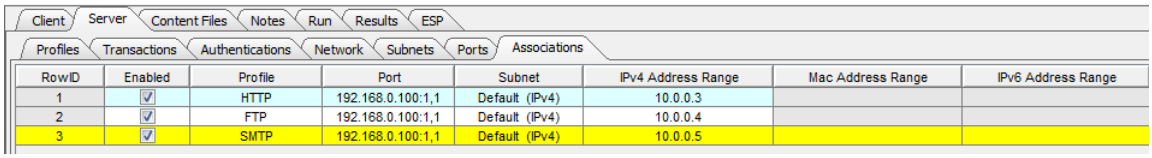
Obrázek D.3: Server - Profiles (profily)

Další upravenou podzáložkou byla Authentiactions. Zde jsem přidal do seznamu uživatelů záznam, kde uživatelské jméno bylo student a heslo také student. Tento záznam můžeme vidět na obrázku D.4.



Obrázek D.4: Záznam v podzáložce Authentications (několik aplikačních protokolů)

Podzáložky Transactions, Network, Subnets a Ports byly totožné s předchozími měřeními. Změna nastává až v záložce Associations, kde jsem přidal 3 záznamy. Každému ze záznamů jsem přidal jeden z vytvořených profilů podzáložky Server - Profiles. Taktéž jsem každému ze záznamů musel nastavit IP adresu podle generovaných akcí z obrázku D.2. Přehled těchto spojení můžeme vidět na obrázku D.5.



Obrázek D.5: Server - Associations (několik aplikačních protokolů)

Zbylé záložky byly nastavené totožně jako u předchozích měření z příloh A, B nebo C.